Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Open source řešení IPsec virtuálních privátních sítí

Open Source Solution of IPsec Virtual Private Networks

2014 **Bc. Radim Daněček**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Radim Daněček**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Open source řešení IPsec virtuálních privátních sítí
Open Source Solution of IPsec Virtual Private Networks

Zásady pro vypracování:

Cílem diplomové práce je návrh, realizace a testování různých řešení IPsec virtuálních privátních sítí s využitím open source softwaru strongSwan.

Osnova práce:
1. Popište IPsec virtuální privátní sítě.
2. Popište software strongSwan.
3. Navrhněte a v laboratorních podmínkách realizujte různé druhy virtuálních privátních sítí s využitím softwaru strongSwan. Ověřte funkčnost navržených řešení.
4. Ověřte možnosti použití certifikátů včetně jejich použití v USB tokenech.
5. Ověřte možnosti použití strongSwan v mobilních telefonech.

Seznam doporučené odborné literatury:

CARMOUCHE James Henry. *IPsec Virtual Private Network Fundamentals*. Indianapolis: Cisco Press, 2006. ISBN 1-58705-207-5.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Petr Machník, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 07.05.2014

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

„Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Dne: 6.5.2014

podpis studenta

Poděkování

Rád bych poděkoval Ing. Petru Machníkovi, Ph.D. za rady, připomínky a odborné vedení při vypracování mé diplomové práce.

Abstrakt

Cílem této diplomové práce je analyzovat problematiku IPsec virtuálních privátních sítí a prakticky realizovat různá řešení za použití softwaru strongSwan. Jsou zde realizovány jednotlivé konfigurace s komentáři a konfiguračními soubory. Všechny realizace jsou doplněny o komentáře a podrobné výpisy sestavených spojení. Dále analyzuje problematiku certifikačních autorit, digitálních certifikátů a jejich využití pro vzájemnou autentizaci. K vytvoření a práci s certifikáty je využit program XCA, který slouží i pro import a využití certifikátů v USB tokenech. Závěrem se práce zabývá využitím VPN klienta strongSwan, v mobilních telefonech s operačním systémem Android.

Klíčová slova

strongSwan, VPN, IPsec, tunelovací, transportní, certifikát, certifikační autorita, digitální podpis, IKE, USB token, Android

Abstract

The aim of this thesis is to analyse the issues concerning IPsec of virtual private nets and put distinct solutions into practise by using the software strongSwan. Each configurations are implemented with comments and configuration files. All implementations are supplemented with remarks and detailed statements of assembled connection. Thereafter, the thesis analyses the issues of certificate authorities, digital certifications and their application in mutual authentication. The programme XCA is utilized for the creation and work with certifications. It functions as import and utilization of certifications in USB token. In closing, the thesis deals with the employment of VPN of strongSwan´s client in mobile phones with the operating system Android.

Key words

strongSwan, VPN, IPsec, tunneling, transport, certificate, certification authority, digital signature, IKE, USB token, Android

Seznam použitých symbolů a zkratek

Zkratka	Anglický význam	Český význam
3DES	Triple Digital Encryption Standard	Bloková šifra
AES	Advanced Encryption Standard	Symetrická bloková šifra
AH	Authentication Header	Autentizace hlavičky
DES	Digital Encryption Standard	Symetrická šifra
EAP	Extensible Authentication Protocol	Autentizační rámec
ESP	Encapsulating Security Payload	Zapouzdření a šifrovaní paketů
GNU	General Public License	Všeobecná veřejná licence
GPLv2	General Public License v2	Všeobecná veřejná licence verze 2
GRE	Generic Routing Encapsulation	Zapouzdření paketů protokolem GRE
HTTP	Hypertext Transfer Protocol	HTTP protokol
IETF	Internet Engineering Task Force	Komise pro techniky používané v Internetu
IKE	Internet Key Exchange	Protokol pro zabezpečení a výměnu klíčů
IP	Internet Protocol	IP protokol
IPsec	IP security	Zabezpečení IP protokolu
IPX	Internetwork Packet Exchange	Síťový protokol, využívaný v operačních systémech Novell NetWare
ISAKMP	Internet Security Association and Key Management Protocol	Rámec pro autentizaci a výměnu klíčů
L2F	Layer 2 Forwarding	Tunelovací protokol spojové vrstvy
L2TP	Layer 2 Tunneling Protocol	Tunelovací protokol
LAN	Local Area Network	Lokální síť
LDAP	Lightweight Directory Access Protocol	Protokol pro ukládání a přístup k datům na adresářovém serveru

MTU	Maximum transmission unit	Maximální přenosová rychlost
NAT	Network Address Translation	Překlad síťových adres
NetBEUI	NetBIOS Extended User Interface	Protokol síťové/transportní vrstvy pro komunikaci v lokální síti
OpenSSL	OpenSSL	Open source implementace protokolů SSL a TLS
OSCP	Offensive Security Certified Professional	Certifikace bezpečnosti
PGP	Pretty Good Privacy	Počítačový program, který umožňuje šifrování a podepisování
PKCS	Public Key Cryptographic Standards	Skupina standardů pro kryptografii
PPTP	Point-to-Point Tunneling Protocol	Způsob realizace virtuální privátní sítě
RSA	Rivest, Shamir, Adleman	Šifra s veřejným klíčem
SA	Security Associations	Bezpečnostní asociace
SPI	Security Parameter Index	Označení bezpečnostní asociace
SSH	Secure Shell	Zabezpečený komunikační protokol
SSL	Secure Sockets Layer	Vrstva bezpečných socketů
TCP	Transmission Control Protocol	TCP protokol
TCP/IP	Transmission Control Protocol/Internet Protocol	TCP/IP protokol
UDP	User Datagram Protocol	UDP protokol
VPN	Virtual Private Network	Virtuálni privátni sít
WAN	Wide Area Network	Rozlehlá síť
Xauth	Xauth	Druhý stupeň ověřování autentizace

Obsah

Úvod

S rozvojem internetu a datových komunikací rostou i nároky na zajištění bezpečnosti a poskytování požadované kvality služeb podle požadavků daného provozu. Proto se čím dál častěji zavádějí VPN sítě, což jsou v podstatě logické sítě v rámci sdílené veřejné infrastruktury. V souvislosti s tím se vynakládají nemalé prostředky na tuto problematiku. Jedním z řešení je software strongSwan, který je šířen pod všeobecnou veřejnou licencí GNU GPLv2 a umožňuje plné využití IPsec protokolu a dalších bezpečnostních prvků pro bezpečnou síťovou komunikaci.

V úvodní kapitole je rozebrána teorie a problematika virtuálních privátních sítí, základní fakta a přehled. Závěrem kapitoly je vysvětlena metodika při zavádění sítí VPN. Třetí kapitola pojednává o VPN sítích s protokolem IPsec. Jsou zde vysvětleny pojmy jako autentizace, digitální certifikát a certifikační autorita. Dále jsou zde objasněny režimy šifrování a informace o doplňujících protokolech, které jsou nezbytné k sestavení a zabezpečení přenosu. Čtvrtá kapitola je teoretickým úvodem k softwaru strongSwan. Objasňuje a vysvětluje jeho vlastnosti a umístění jeho konfiguračních souborů v systému Linux.

Pátá kapitola se podrobněji věnuje praktickým využitím různých konfigurací VPN sítí s využitím strongSwan. U každého řešení jsou uvedeny výpisy sestaveného spojení a zachycení provozu programem Wireshark. V úvodu kapitoly je podrobněji vysvětlen popis a práce s konfiguračními soubory. Následují tři síťová řešení (host-host, vzdálený klient, site-to-site), kdy autentizace probíhá na základě předsdíleného klíče. Další část kapitoly se věnuje konfiguracím, kdy pro autentizaci je využito certifikátů, které jsou digitálně podepsány certifikační autoritou. Pro tento účel je využit program XCA, jehož jádro tvoří knihovna OpenSSL. Těchto certifikátů je poté využito pro otestování konfigurací spojení dvou hostů. Následně jedno z nejvyužívanějších spojení, kdy se vzdálený klient připojuje do vnitřní sítě za použití klientského certifikátu. Následuje řešení IPsec a NAT, kdy PC s Linuxem simuluje NAT směrovač. Další kapitola se teoreticky zabývá použití certifikátů v UBS tokenech. Poslední kapitola ukazuje praktické využití strongSwan VPN klienta pro mobilní telefony, kdy je realizováno spojení tohoto Android klienta na bránu, kde běží strongSwan. Pro ověření uživatele je využito certifikátů a protokolu EAP-MSCHAPv2, kdy se klient přihlašuje pod nastaveným uživatelským jménem a heslem. U všech realizací jsou okomentovány konfigurační soubory.

1 Virtuální privátní sítě (VPN)

V dnešním propojeném světě je často zapotřebí přesouvat informace z jednoho stanoviště na druhé. Ať už se jedná o přenos na konec města nebo přes celou zeměkouli, základním problém zůstává stále stejný: Jak můžeme bezpečně transportovat naše data? Pro mnoho let byly tyto transporty prováděny drahými soukromými linkami, které „pronajímali“ prodejci komunikací, takže společnosti měly „soukromý (privátní)“ segment pro takové komunikace. Čím větší byla vzdálenost, tím dražší vyla tato připojení, díky čemuž se sítě WAN staly přepychem, který si mnohé společnosti nemohli dovolit. Zároveň si ale v té době mnohé firmy nemohly dovolit bez těchto sítí existovat. A tak jak se postupně stávalo širokopásmové připojení k Internetu dostupné pro více firem, tak se koncept používání existující struktury Internetu jakožto kabeláže WAN začal zdát nezajímavý. Náklady mohly být výrazně sníženy použitím tehdy již dostupných veřejně přístupných bodů. Hlavním problémem zůstávalo, jak udržet data zabezpečená. Protože sdílíme světovou „skupinovou přípojku“ s kýmkoli dalším, kdo se připojuje k Internetu, jak můžeme zabezpečit, že jsou naše data ochráněna před slídily? Řešením je technologie virtuálních privátních sítí (Virtual Private Network – VPN).

1.1 Základní fakta o VPN

Síť VPN (někdy také nazývaná „tunel“ VPN) je v podstatě připojení, které je pomocí šifrovacích nebo autentizačních technologií zavedeno nad existující veřejnou nebo sdílenou infrastrukturou tak, aby byl zabezpečen užitečný obsah připojení. Tím se vytvoří „virtuální“ segment mezi jakýmikoliv dvěma entitami, které k sobě mají přístup. VPN lze vytvářet přes sdílenou infrastrukturu místní sítě (LAN), přes WAN připojení nebo přes Internet.

Sítě VPN můžeme rozdělit do tří základních typů podle nastavení: hostitel-hostitel, hostitel-vstupní brána (gateway) a vstupní brána-vstupní brána (gateway-gateway). Pro VPN, které prochází Internetem, by bylo možno využívat kterýkoliv z těchto scénářů, i když VPN typu hostitel-hostitel se také používá jako způsob pro soukromou komunikaci v rámci segmentů místní sítě. Bez ohledu na to jaký typ VPN se používá, jaký typ konfigurace daná síť VPN reprezentuje nebo jakou sdílenou infrastrukturou síť VPN prochází, je síť VPN mocným nástrojem, který můžeme použít mnoha způsoby pro vytvoření bezpečného komunikačního kanálu.

1.2 Přehled sítí VPN

Virtuální privátní síť (Virtual Private Network, VPN) je šifrované síťové spojení, které při své činnosti využívá mezi dvěma koncovými body bezpečný komunikační tunel, vedený po Internetu či jiné síti WAN. V síti VPN se tak vytáčená připojení vzdálených uživatelů a pronajaté linky nebo okruhy Frame Relay do vzdálených pracovišť nahrazují za místní připojení k poskytovateli internetových služeb či jinému přístupovému bodu. Díky stále běžnějšímu širokopásmovému připojení k Internetu je toto řešení stále dostupnější i pro malé pobočky a domácnosti. Kromě prvotní investice do sítě VPN je již připojení dalších pracovišť a uživatelů za minimální cenu.

Nad sítí VPN může každý vzdálený uživatel bezpečně a spolehlivě komunikovat s privátní sítí LAN i přes veřejný Internet. Také rozšiřování o další uživatele a pracoviště je v síti VPN podstatně snazší než u pronajaté linky. Uvedená škálovatelnost neboli možnost rozšíření je fakticky největší výhodou VPN proti klasické pronajaté lince. Navíc, zatímco náklady na pronajaté linky jsou úměrné vzdálenosti mezi pracovišti, nehraje u sítě VPN prostorové rozmístění prakticky žádnou roli.

Virtuální privátní síť pomocí šifrování IPsec bezpečně rozšiřuje dosah privátního intranetu, a to nad běžným Internetem či jinou síťovou službou. Umožňuje tak bezpečné vedení elektronické komerce a extranetových spojení s mobilními zaměstnanci, obchodními partnery, dodavateli a zákazníky. Rozeznáváme tři základní typy sítí VPN:

VPN pro vzdálený přístup. Umožňuje bezpečně připojení jednotlivých vytáčených uživatelů k centrálnímu pracovišti přes Internet nebo jinou veřejnou síťovou službu. Tento typ sítě VPN představuje spojení uživatele a sítě LAN, a slouží tak zaměstnancům pro připojení z terénu. Na počítači zaměstnance běží speciální klientský software VPN, který zavádí bezpečnou linku do podnikové sítě LAN.

VPN pro spojení pracovišť. Tyto sítě rozšiřují stávající LAN ve firmě do dalších budov a pracovišť pomocí určitého specializovaného vybavení; vzdálení zaměstnanci pak mohou využívat stejné síťové služby jako pracovníci v ústředí. Uvedený typ sítě VPN bývá trvale aktivně propojený; někdy se označují také jako hardwarové VPN, intranetové VPN nebo jako VPN mezi sítěmi LAN (LAN-to-LAN, site-to-site).

Extranetové VPN. Poslední typ poskytuje bezpečné spojení s obchodními partnery, dodavateli a zákazníky za účelem vedení elektronické komerce. Extranetové sítě VPN jsou rozšířením intranetových VPN a vnitřní síť je v nich navíc chráněna pomocí firewallů. Dobrým příkladem může být firma, která natolik úzce spolupracuje s dodavateli a partnery, že vzájemné potřeby zajišťuje přímým propojením informačních systémů – je to tedy vyšší forma dodavatelsko-odběratelských vztahů. Extranet umožňuje rychlejší výměnu informací.

1.3 Základní metodika při zavádění sítí VPN

Základní koncept, na kterém stojí technologie VPN, spočívá ve vytváření bezpečného komunikačního kanálu pomocí šifrování. Komunikace může být zabezpečena šifrováním na mnoha odlišných vrstvách síťového modelu, například na vrstvě:

- Aplikační
- Transportní
- Síťové
- Spojové

Na aplikační vrstvě může být šifrování zajištěno programově, například šifrovací metodou Pretty Good Privacy (PGP), nebo pomocí zabezpečených kanálů, jako je například Secure Shell (SSH). Navíc lze použít programy pro jedinou relaci, jako například pcAnywhere, nebo programy, jako je například Terminal Server, společně se šifrováním a vytvořit tak chráněné vzdálené komunikace. Většina těchto programů pracuje z hostitelského počítače na hostitelský počítač, což znamená, že nabízejí ochranu obsahu paketů, a nikoliv již paketu samotného. Výjimkou je v tomto ohledu SSH, které lze použít v režimu port-forwarding pro vytvoření „tunelu“.

Co se týká transportní vrstvy, lze zde použít protokoly, jako je například Secure Sockets Layer (SSL), pro ochranu užitečného obsahu specifické komunikace mezi dvěma stranami. Typicky se tento způsob zabezpečení používá při komunikaci s webovým prohlížečem. Opět ale platí, že jsou chráněny pouze obsahy komunikace (tedy užitečné obsahy paketů), ale IP pakety, které tyto informace obsahují, může kdokoliv získat.

V síťové vrstvě již protokoly, jako je například IPsec, nešifrují pouze užitečný obsah paketu, ale šifrují také TCP/IP informace. I když jsou informace o IP adrese nutné pro správné směrování paketu, ostatní informace z vyšší vrstvy, jako je např. typ transportních protokolů a asociovaných portů, lze kompletně zašifrovat. Pokud zařízení přenosové brány (gateway), jako je například směrovač, firewall nebo sdružovač, provádí šifrování v konceptu „tunelování“ je možné v paketu skrýt také IP adresu koncové stanice.

Tunelovací protokol druhé vrstvy (Layer 2 Tunneling Protocol – L2TP) a (Point-to-Point Tunneling Protocol – PPTP) a umožňuje šifrování paketů zaslaných přes PPTP na spojové vrstvě (druhá vrstva).

Navzdory skutečnosti, že se tyto šifrovací technologie vyskytují v mnoha odlišných síťových vrstvách, mohou být všechny také součástí sítí VPN. Některé z nich nicméně nemusí být schopny zacházet se všemi povinnostmi sítí VPN, takže je nutno je podpořit pomocí dalších aplikací nebo protokolů.

Při vytváření privátní sítě v prostředí veřejného Internetu se technologie VPN opírá o takzvané tunelování. V podstatě to znamená, že systém vezme celý paket dat a zapouzdří jej do jiného paketu, který následně přenese po síti; tato síť musí pouze znát protokol vnějšího paketu, jehož prostřednictvím data vstupují a vystupují ze sítě. Do celého tunelování jsou zapojeny tři různé protokoly.

- **Přenášený (nesený) protokol**

Původní datový protokol, obvykle IP, který se má zašifrovat pro přenos v síti VPN. Podle potřeby je možné přenášet i jiné protokoly, například IPX a NetBEUI.

- **Zapouzdření – obalový protokol**

Tento protokol (GRE, IPsec, L2F, PPTP, L2TP) se „obalí“ okolo původních dat; říkáme, že data jsou v něm zapouzdřena. V současné době je de facto standardem pro zapouzdření dat protokol IPsec; zapouzdření umožňuje šifrování a ochranu celého přenášeného paketu. Pro správnou činnost tunelu musí obě jeho rozhraní podporovat protokol IPsec.

- **Nosný protokol**

Protokol, přes který v síti putují informace. Původní přenášený paket se zapouzdří v obalovém protokolu; výsledný paket se dále doplní o hlavičku nosného protokolu (je jim obvykle IP) a konečně se přenese po síti.

Tunelování pracuje v sítích VPN velice dobře, protože do paketu IP můžeme zapouzdřit a bezpečně přenést i protokoly, které nejsou na Internetu podporovány. Na začátku celého tunelovaného přenosu v sítí VPN se totiž datový paket ze zdrojové sítě LAN zapouzdří, neboli obalí do nové hlavičky, podle níž ho správně rozpoznají a doručí všechny mezilehlé sítě. Po dokončení přenosu se hlavička tunelovacího protokolu odřízne a znovu se obnoví původní paket, který se konečně předá k doručení do cílové LAN.

Mechanismus tunelování umožňuje přenos dat i po cizích sítích, sám o sobě ale ještě soukromí nezajišťuje. Pro zabezpečení tunelovaného přenosu dat proti odposlechu a falšování je nutné veškerý provoz v síti VPN šifrovat. Navíc, sítě VPN obsahují zpravidla i další komponenty, jako jsou firewally na obvodu sítě.

Ve VPN pro spojení pracovišť se v roli obalového protokolu používá obvykle IPsec, případně obecný protokol GRE (Generic Routing Encapsulation). Protokol GRE obsahuje informace o typu zapouzdřeného paketu a o spojení mezi klientem a serverem. Rozdíl mezi oběma protokoly spočívá v úrovni zabezpečení – IPsec je bezpečnější, zatímco GRE tuneluje nejen pakety IP, ale i jiných protokolů. Při odeslání jiných protokolů než IP (například IPX) přes tunel je nejlépe oba protokoly (IPsec a GRE) použít společně. **Při zpracování této kapitoly jsem čerpal z [1] a [8].**

2 Přehled sítí VPN s protokolem IPsec

Standardem pro vytváření virtuálních sítí VPN se stal protokol IPsec. Jeho implementaci nabízí hned několik různých výrobců; protože je definován v dokumentu RFC ze sdružení IETF (Internet Engineering Task Force), je zajištěna i vzájemná spolupráce zařízení od různých výrobců. Protokol IPsec nabízí standardní prostředky pro navázání autentizačních a šifrovacích služeb mezi komunikujícími partnery. Pro účely tohoto výkladu budeme za partnery IPsec (peers) považovat zařízení, která se nacházejí na obou koncích tunelu VPN. Z pohledu referenčního modelu OSI tvoří IPsec síťovou vrstvu, přičemž zajišťuje ochranu a autentizaci paketů IP mezi zúčastněnými zařízeními IPsec (neboli partnery), jako jsou směrovače a firewally Cisco. Protokol IPsec zajišťuje v sítích následující bezpečnostní funkce:

- **Důvěrnost dat.** Odesílatel IPsec může data před přenosem po síti zašifrovat; pokud je hacker neumí dešifrovat, jsou mu k ničemu.
- **Integrita dat.** Přijímající koncový bod IPsec autentizuje veškeré pakety od odesílatele a kontroluje tak, jestli nebyla data při přenosu pozměněna.
- **Autentizace původu dat.** Příjemce IPsec může dále autentizovat zdroj odeslaných paketů IPsec; tato služba je závislá na službě integrity dat.
- **Ochrana proti opakování relace.** Příjemce IPsec může detekovat opakovaně pakety a zamítnout je.

Protokol IPsec tak chrání citlivá data při přenosu v nechráněných sítích, přičemž jeho bezpečnostní služby pracují na síťové vrstvě jednotlivé pracovní stanice, osobní počítače či aplikace tak nemusíme zvlášť konfigurovat. Tato výhoda znamená současně i velkou úsporu nákladů; nemusíme zajišťovat množství bezpečnostních služeb, které fakticky nepotřebujeme, a koordinovat zabezpečení zvlášť pro každou aplikaci a pro každý počítač, ale namísto toho změníme přímo síťovou infrastrukturu, která bude nově zajišťovat všechny potřebné bezpečnostní služby. Díky tomu je možné řešení IPsec snadno implementovat do středních až velkých, nebo rostoucích sítí, kde je často nutné bezpečně propojit mnoho různých zařízení.

Součástí IPsec jsou zdokonalené bezpečnostní funkce, jako například vylepšené šifrovací algoritmy a obecnější autentizace. Podnikové sítě připojené k Internetu tak mohou s protokolem IPsec snadno postavit flexibilní a bezpečný přístup k síti VPN. S technologií IPsec mohou zákazníci nad veřejným Internetem vybudovat síť VPN s bezpečným šifrováním a ochranou proti odčerpání dat z kabelů, proti odposlechu a jiným útokům vedeným vůči privátní komunikaci.

Protokol IPsec mohou využívat jen takové systémy, které jej podporují. Všechna zařízení musí navíc používat jeden stejný klíč a firewally v jednotlivých sítích musí nastavené podobné zásady zabezpečení.

Jako ochranu proti neoprávněnému prohlížení či modifikaci dat v síti a jako ochranu dat při přenosu po nechráněné síti (například po veřejném Internetu) poskytuje IPsec služby autentizace a šifrování. Protokol IPsec může šifrovat data mezi dvojicemi nejrůznějších zařízení, například:

- Směrovač a směrovač
- Firewall a směrovač
- Firewall a firewall
- Uživatel a směrovač
- Uživatel a firewall
- Uživatel a koncentrátor VPN
- Uživatel a server

Samotný IPsec představuje soubor otevřených standardů, definovaných sdružením IETF. Tento soubor protokolů zajišťuje bezpečnost přenosu citlivých informací přes nechráněné sítě, jakou je Internet.

2.1 Autentizace a integrita dat

Při navazování vztahu důvěry slouží autentizace k ověření totožnosti (identity) obou koncových bodů sítě VPN a uživatelů, kteří přes ni odesílají provoz. Koncovým bodem může být přitom klient VPN, koncentrátor VPN, firewall nebo směrovač. Autentizace je proces protokolu IPsec, který nastupuje po šifrování dat u odesílatele a před jejich dešifrováním na straně příjemce; je nezbytnou součástí IPsec, protože kontroluje, že odesílatel i příjemce jsou skutečně tím, za koho se vydávají.

Další funkcí souboru protokolů IPsec je integrita dat (celistvost), která znamená, že přijatý paket nebyl během přenosu nijak pozměněn. Tato vlastnost se zajišťuje pomocí jednosměrného hashovacího algoritmu (otisku zprávy), který je v podstatě ekvivalentem šifrovaného kontrolního součtu. Odesílající strana paket zašifruje a autentizuje, a poté nad jeho celým zněním spustí jednosměrný hash. Zajímavou vlastností hashe je jeho pevná velikost, můžeme říci, že je to také jistý bezpečnostní mechanismus, protože hacker nemá šanci se takto dozvědět velikost vstupního pole. Šifrované pole s jednosměrným hashem se připojí ke zprávě a odešle. Přijímající strana odebere hash, z „čisté“ zprávy si vypočte vlastní hodnotu a obě porovná. Hash se vypočítává nad veličinami stejného paketu – čas odeslání, počet bajtů atd. – takže pokud zpráva nebyla po cestě pozměněna, musí se výsledky shodovat. Jestliže se výsledky liší, paket se jako vadný zahodí a IPsec provede novou dohodu o bezpečnostních parametrech.

2.2 Digitální podpis

Digitální podpis slouží k autentizaci zdroje dat a ke kontrole integrity přenášených dat. Digitální podpis využívá kombinaci hashovacího algoritmu a asymetrické šifry.

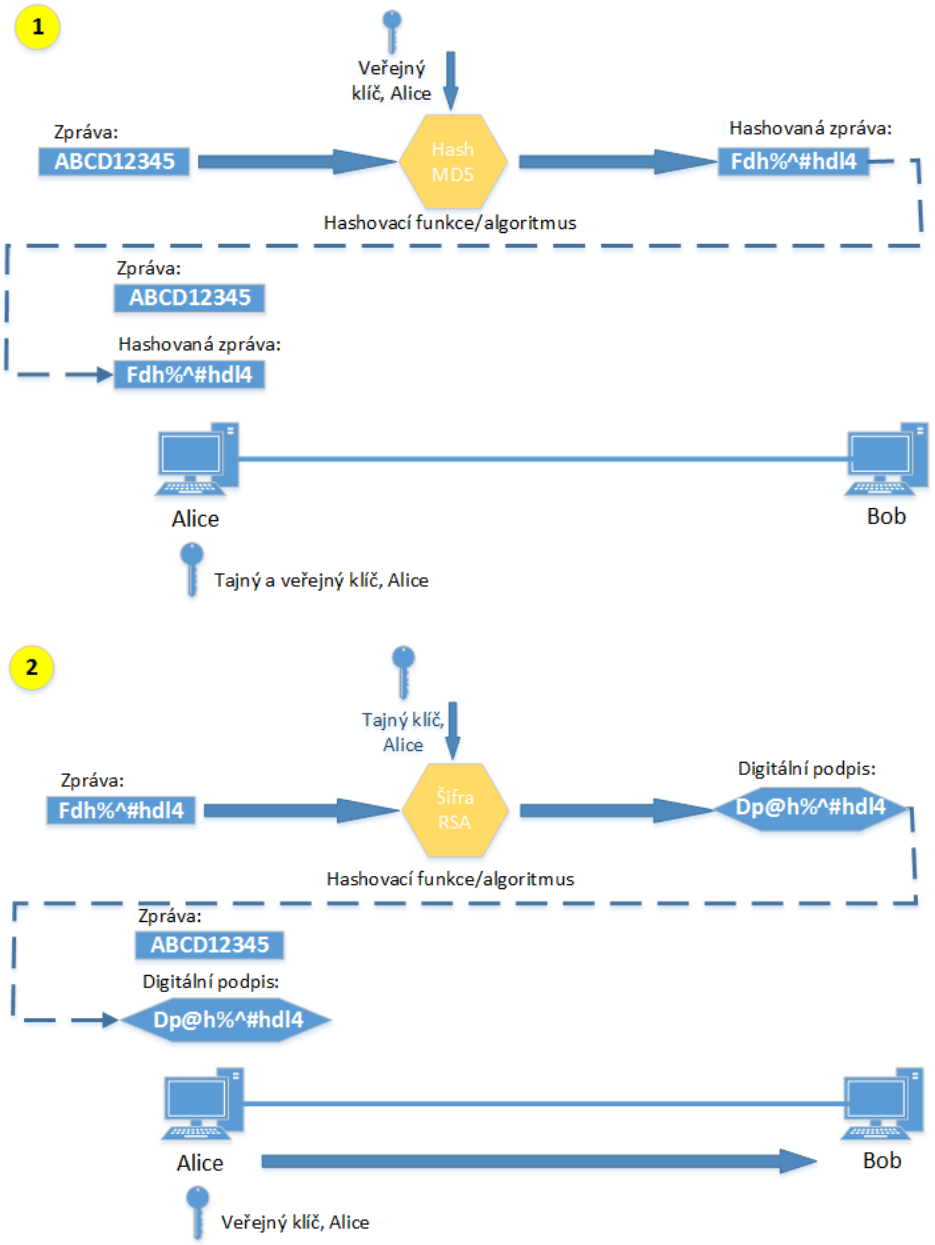
Vše je názorně ukázáno na následujícím příkladu, obrázek 3.1:

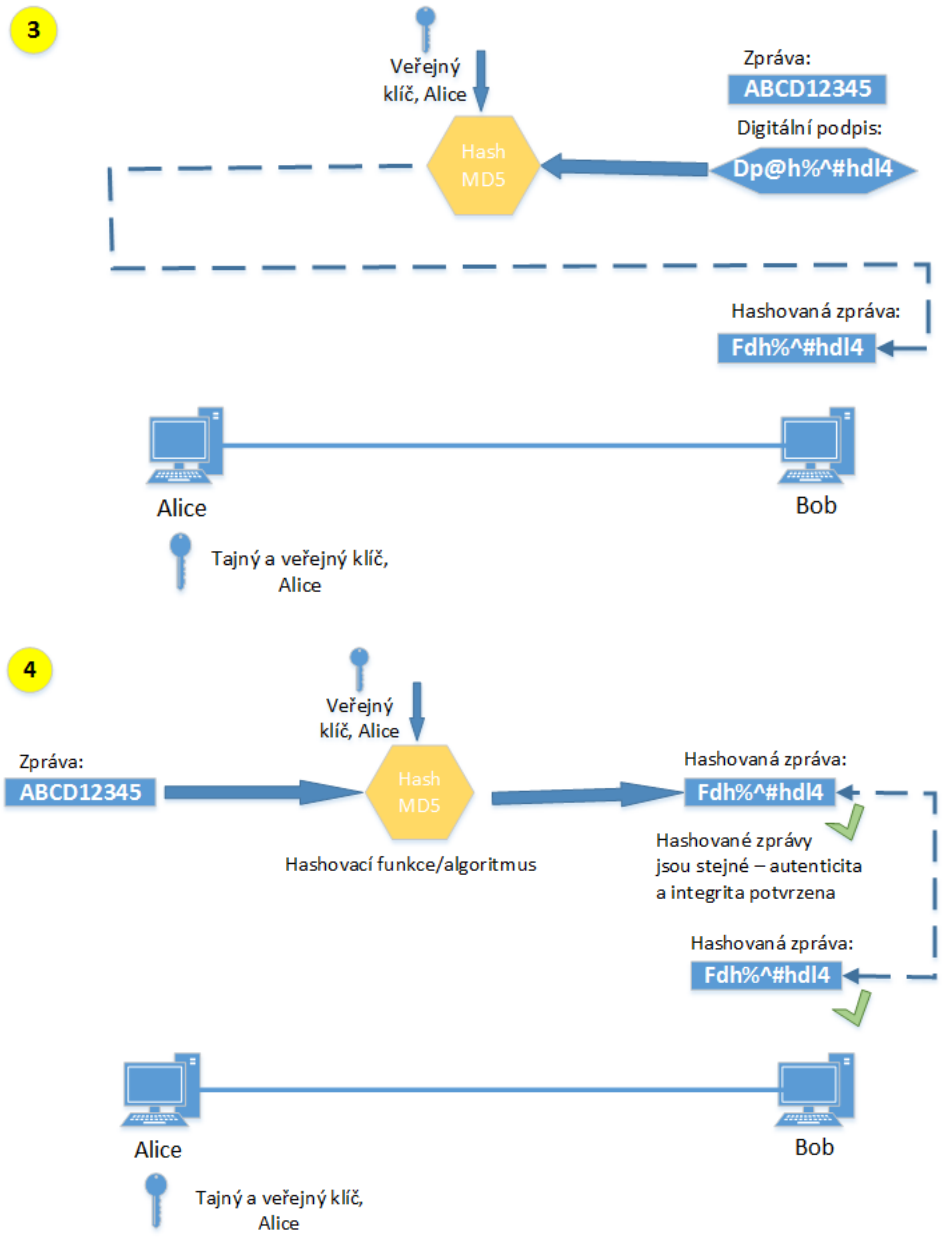
1) Alice si vygeneruje tajný a veřejný klíč. Veřejný klíč pošle Bobovi. Pomocí veřejného klíče zahešuje zprávu, kterou chce poslat Bobovi.

2) Zahashovanou zprávu Alice následně zašifruje asymetrickou šifrou s využitím svého tajného klíče, čímž vznikne digitální podpis. Ten je připojen k původní zprávě a spolu s ní přenesen k Bobovi.

3) Bob dešifruje pomocí Alicina veřejného klíče přijatý digitální podpis, čímž získá zahešovanou zprávu.

4) Pomocí Alicina veřejného klíče zahashuje Bob přijatou zprávu. Výsledek porovná s výsledkem dešifrování v kroku 3. Pokud jsou obě hodnoty stejné, považuje Bob autenticitu Alice za prokázanou – pouze ona má tajný klíč, kterým mohla hashovanou zprávu zašifrovat (což je opačný postup než u běžného asymetrického šifrování). Současně je hešováním ověřena integrita zprávy. Problémem ovšem zůstává riziko podvržení Alicina veřejného klíče útočníkem, který by se za Alici vydával. Samotnou zprávu je samozřejmě současně možné zabezpečit asymetrickým šifrováním – zašifrovat Bobovým veřejným klíčem a dešifrovat Bobovým tajným klíčem.


1
Veřejný klíč, Alice
Zpráva:
ABCD12345
Hash MD5
Hashovaná zpráva:
Fdh%^#hdl4
Hashovací funkce/algoritmus
Zpráva:
ABCD12345
Hashovaná zpráva:
Fdh%^#hdl4
Alice
Bob
Tajný a veřejný klíč, Alice
2
Tajný klíč, Alice
Zpráva:
Fdh%^#hdl4
Šifra RSA
Digitální podpis:
Dp@h%^#hdl4
Hashovací funkce/algoritmus
Zpráva:
ABCD12345
Digitální podpis:
Dp@h%^#hdl4
Alice
Bob
Veřejný klíč, Alice



Obrázek 2.1: Ukázka použití digitálního podpisu

2.3 Certifikační autorita

Ačkoli použití šifrování a digitálních podpisů dokáže poměrně spolehlivě zabezpečit přenášená data, ověřit autenticitu zdroje dat a integritu těchto dat, zůstává problémem bezpečná distribuce velkého množství veřejných klíčů pro velké množství komunikujících stran. Řešením je vytvoření certifikační autority, jejímž úkolem je ověření původu těchto veřejných klíčů. V takovém případě je potřeba bezpečně doručit pouze jeden veřejný klíč – veřejný klíč certifikační autority (aby i on nemohl být podvržen). Funkce certifikační autority je názorně ukázána na následujícím příkladu:

1) Alice a Bob si vyžádají kořenový certifikát, tj. certifikát certifikační autority, který obsahuje veřejný klíč certifikační autority. Navíc může ještě proběhnout ověření autenticity certifikační autority.

2) Alice a Bob se zaregistrují u certifikační autority a pošlou jí své veřejné klíče k ověření.

3) Certifikační autorita digitálně podepíše certifikáty obsahující tyto veřejné klíče pomocí svého tajného klíče.

4) Certifikační autorita pošle Alici a Bobovi jejich certifikáty obsahující jejich veřejný klíč, digitální podpis certifikátu, platnost certifikátu, údaje o vydavateli certifikátu a některé další údaje. Tyto certifikáty si Alice a Bob uloží pro pozdější použití.

5) Pokud chtějí Alice a Bob spolu komunikovat, vymění si navzájem své certifikáty s digitálním podpisem.

6) Alice a Bob ověří autenticitu toho druhého tím, že ověří digitální podpis přijatého certifikátu. To provedou pomocí veřejného klíče certifikační autority (viz. krok 1).

7) Nyní můžou Alice i Bob používat veřejný klíč od toho druhého k šifrování posílaných dat. Své tajné klíče použijí k dešifrování přijatých dat. V praxi (např. u protokolu IPsec) se asymetrické šifrování použije jen k výměně klíče symetrické šifry. Užitečná data, která mají být při svém přenosu zabezpečena, se pak šifrují pomocí této symetrické šifry.

2.4 Obsah certifikátu

Jedná se o datový soubor, uložený ve standardním, mezinárodně platném formátu. Pro certifikáty se používá mezinárodní norma X.509, která jednoznačně popisuje strukturu certifikátu. Každý certifikát musí obsahovat následující údaje:

1) Sériové číslo - je unikátní, žádná certifikační autorita nesmí vydat dva certifikáty se stejným číslem

2) Datum počátku a konce platnosti certifikátu. Doba platnosti certifikátu zpravidla souvisí s tím, k jak silnému klíči je certifikát vydán. Nejběžnější doba platnosti certifikátu je jeden rok.

3) Identifikační údaje subjektu, kterému je certifikát vydán. Tyto údaje si certifikační autorita musí spolehlivě ověřit, například v případě osobního certifikátu kontrolou dokladu totožnosti.

4) Veřejný klíč. Nejčastěji se používá délka klíče 1024 bitů. Kromě klíče, certifikát obsahuje druh algoritmu, který bude pro podepisování používán.

5) Identifikační údaje vydavatele certifikátu, tedy certifikační autority.

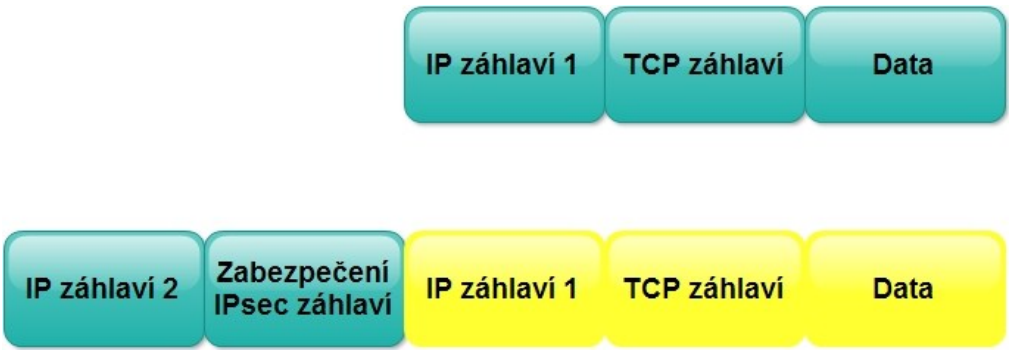
2.5 Režimy šifrování

K zabezpečení přenosu multimediálních lze použít sadu protokolů IP security (IPsec), tyto protokoly zabezpečují komunikaci na úrovni síťové vrstvy, tj. jednotlivé datagramy. Jsou nezávislé na protokolech vyšších vrstev TCP/UDP. IPsec vznikl jako součást IPv6 a později byl poté i implementován pro IPv4. IPsec definuje dva režimy zabezpečení IP datagramů:

- Transportní režim
- Tunelovací režim

2.5.1 Tunelovací režim

Tunelovací režim chrání celý datagram, který je zapouzdřen. Původní paket se zapouzdří a vloží do nového IP datagramu, v němž za novým IP záhlavím následuje bezpečnostní záhlaví. V případě tunelovacího režimu tak existují dvě IP záhlaví, vnitřní (původní – IP záhlaví 1) a vnější (nové – IP záhlaví 2). Tunelovací režim primárně chrání provoz mezi sítěmi s nedůvěryhodnou cestu. Používá se tehdy, když je potřeba propojit celé sítě. Princip zabezpečení tunelovacím režimem je zobrazen na obrázku 3.2, žlutá barva značí zabezpečenou část IP datagramu.



Obrázek 2.2: Tunelovací režim

2.5.2 Transportní režim

Je z těchto dvou režimů méně bezpečný. Šifruje pouze datovou část IP datagramu. IP datagram je rozšířen o bezpečnostní záhlaví, které je vloženo mezi záhlaví IP datagramu a záhlaví vyšší vrstvy (TCP/UDP). Bezpečnostní hlavička pak určuje, jak je datová část IP datagramu zabezpečena. IPsec v transportním režimu se používá k ochraně komunikace v rámci jedné sítě. Princip zabezpečení transportním režimem je zobrazen na obrázku 3.3, žlutá barva značí zabezpečenou část IP datagramu.

Obrázek 2.3: Transportní režim

2.6 Protokoly IPsec

Balík IPsec používá vzájemně se doplňující protokoly, které společně tvoří ucelený a bezpečný systém standardů, jež se ideálně hodí i pro sítě VPN. Podívejme se na ně podrobněji:

- **ISAKMP (Internet Security Association and Key Management Protocol).** Popisuje fázi dohody o spojení v IPsec, ve kterém se navazuje spojení VPN; protokol Oakley pak definuje metodu navázání výměny autentizovaného klíče. Uvedená metoda může mít několik různých režimů činnosti a výchozí údaje pro klíče („klíčový materiál“) může odvozovat pomocí speciálních algoritmů jako je Diffie_Hellman. Součástí protokolu ISAKMP je také standard IKE (Internet Key Exchange), jenž definuje postup pro dohodu bezpečnostních parametrů (například doba života bezpečnostních asociací SA, typ šifrování atd.) a pro potvrzení věrohodnosti klíčů. ISAKMP používá UDP port 500.
- **ESP (Encapsulating Security Payload).** Zajišťuje důvěrnost a ochranu dat s volitelnými službami autentizace a detekce opakování relace. Protokol ESP definuje plné zapouzdření uživatelských dat; můžeme jej používat samostatně, nebo ve spojení s AH. Běží nad protokolem TCP s porty 50 a 51 a blíže je popsán v dokumentu RFC 2406.
- **AH (Authentication Header).** Poskytuje autentizaci a volitelně také ochranu proti opakování relace. Jeho služby jsou však omezené jen na část hlavičky IP a rozšířené hlavičky, ale nezajišťuje již šifrování dat – pomocí jednosměrného haše vytváří z paketu pouze otisk zprávy. Autentizační hlavička AH se vkládá do samostatných chráněných dat (například do úplného datagramu IP). Protokol AH můžeme používat samostatně, nebo ve spojení s ESP, kterým byl dnes již ale v podstatě překonán. podrobněji je AH popsán v RFC 2402.

2.7 Bezpečnostní asociace SA

Bezpečnostní asociace (Security Association SA) zavádějí vztah důvěry mezi dvěma partnerskými zařízeními a koncové body sítě VPN se tak pomocí nich mohou dohodnout na množině přenosových pravidel, kdy si vzájemně potvrzují takzvané zásady (politiky) s potenciálním partnerem. Bezpečnostní asociaci tak můžeme považovat za jakousi „smlouvu“, v nichž se potvrdí a závazně nastaví různé parametry navazovaného spojení.

Určitou bezpečnostní asociaci popisuje IP adresa, identifikátor bezpečnostního protokolu a jedinečná hodnota indexu bezpečnostního parametru SPI (security parameter index). Index SPI je přitom 32bitové číslo, zapisované do hlavičky paketu. Rozlišujeme dva typy bezpečnostních asociací:

- **IKE (Internet Key Exchange).** Zajišťuje dohodu klíčů, autentizaci partnerů, správu klíčů a jejich výměnu. Je to obousměrný protokol a jako takový vytváří bezpečný komunikační kanál mezi oběma zařízeními, která se dohodnou na šifrovacím algoritmu, hašovacím algoritmu, autentizační metodě a případných informacích o skupině. Výměna klíčů zde vychází z algoritmu Diffie-Hellman, přičemž síťoví administrátoři mohou IKE úzce svázat se systémy pro správu zásad. Jako ochrana proti útoku in the middle (kdy útočník odposlouchává pakety v síti, upravuje je a v pozměněné podobě je vkládá zpět do komunikačního proudu) slouží zdokonalená verze algoritmu Diffie-Hellman s označením STS (Station To Station); tento protokol provádí vzájemnou autentizaci obou zařízení pomocí digitálních podpisů a certifikátů s veřejným klíčem.
- **IPsec SA (bezpečnostní asociace IPsec).** Dvě zařízení, která chtějí vytvořit IPsec tunel, se musí dohodnout na řadě parametrů. Toto dohadování má na starosti IPsec SA.

  Dohadovanými parametry jsou například:

  - Mód činnosti – transportní nebo tunelovací.
  - Způsob zapouzdření paketů – protokol ESP nebo AH, druh symetrické šifry k zašifrování dat (DES, 3DES, AES).
  - Oba konce tunelu (peer) – za předpokladu, že nejde o dynamicky vytvářený tunel.
  - Provoz, který se má zabezpečit – provoz, který má být zašifrován na jednom konci musí odpovídat provozu, který se má dešifrovat na druhém konci.
  - MTU (Maximum Transfer Unit) v rámci tunelu.
  - SPI (Security Parameter Index).
  - Doba trvání IPsec SA.

Bezpečnostní asociace se využívají v protokolech IKE i IPsec, i když obě asociace jsou pak vzájemně nezávislé. Asociace IPsec SA jsou navíc jednosměrné a v každém z bezpečnostních

protokolů (AH a/nebo ESP) jsou jedinečné. Úlohou bezpečnostních asociací je definovat, které protokoly a algoritmy se budou aplikovat na citlivé pakety a jaké výchozí údaje pro klíče se budou mezi oběma partnery používat. Asociace IPsec SA je možné zavést dvěma způsoby:

Ruční bezpečnostní asociace s předem sdílenými klíči. Neprobíhá zde žádná dohoda o asociacích, takže oba systémy musí mít stejnou konfiguraci, jinak se síťový provoz pod IPsec nemůže správně přenášet. Ruční konfigurace není nijak obtížná; na druhé straně se ale předem sdílené klíče dosti těžko mění, protože tunel po jednostranné změně přestane fungovat – předem sdílené klíče se proto obvykle nikdy nemění.

Asociace zaváděné pomocí IKE. Jestliže bezpečnostní asociace zavádíme pomocí protokolu IKE, mohou se oba partneři na parametrech nových bezpečnostních asociací dohodnout. To znamená, že určíme seznam povolených parametrů (například přípustných transformací).

2.8 Diffie-Hellmanův algoritmus

Diffie-Hellmanův algoritmus slouží k vytvoření a bezpečné výměně sdíleného tajného klíče, který bude použit pro symetrické šifrování přenášených dat. V praxi se používají 3 varianty tohoto algoritmu (group 1, group 2, group 5), které jsou schopny vytvářet různě dlouhé tajné klíče. Například pro potřeby nejbezpečnější symetrické šifry AES je třeba vytvořit velmi dlouhý klíč, k čemuž se použije Diffie-Hellman group 5.

Diffie-Hellmanův algoritmus funguje následujícím způsobem:

1) Bob (resp. Alice) vygeneruje dvě náhodná vysoká prvočísla P a Q. Ty pak pošle Alici (resp. Bobovi).

2) Alice vygeneruje náhodné vysoké číslo A a s jeho pomocí vypočte hodnotu A*: A* = QA mod P. Hodnotu A* pošle Bobovi.

3) Bob vygeneruje náhodné vysoké číslo B a s jeho pomocí vypočte hodnotu B*: B* = QB mod P. Hodnotu B*pošle Alici.

4) Alice a Bob odvodí hodnotu sdíleného tajného klíče z hodnot B*, resp. A* podle těchto rovnic: K = (B*)A mod P (Alice), K = (A*)B mod P (Bob).

5) Nyní mají Alice i Bob společný tajný klíč, který mohou použít k symetrickému šifrování přenášených dat.

Pozn.: mod (modulo) – zbytek po celočíselném dělení.

I komunikace přes IKE SA spojení může být zabezpečena klíčem vytvořeným pomocí tohoto algoritmu.

2.9 Protokol IKE - podrobnější popis

Pro účely dosažení dohody o zabezpečení mezi dvěma počítači vytvořilo sdružení IETF (Internet Engineering Task Force) standardní metodu přidružení zabezpečení a rozpoznání výměny klíčů nazvanou IKE (Internet Key Exchange). Tento protokol provádí následující akce:

- centralizuje správu přidružení zabezpečení, čímž dochází ke zkrácení času potřebného k připojení,
- generuje a spravuje sdílené tajné klíče, které se používají k zabezpečení informací.

Díky tomuto procesu je chráněna nejen komunikace mezi počítači, ale také vzdálené počítače, které požadují zabezpečený přístup do podnikové sítě. Proces probíhá také v případech, kdy vyjednávání s koncovým cílovým počítačem (s koncovým bodem) probíhá prostřednictvím bezpečnostní brány.

2.9.1 IKE fáze 1

IKE fáze 1 zahrnuje proces vytváření IKE SA (někdy označováno i jako ISAKMP SA), tj. proces vytváření zabezpečeného spojení IKE, které se dále využije ve fázi 2.

Existují dvě varianty této fáze – **hlavní mód (main mode)** a **agresivní mód (aggressive mode)**. V hlavním módu proběhne třikrát výměna zpráv mezi oběma stranami, v agresivním módu jen dvakrát. Agresivní mód je tedy rychlejší a méně výpočetně náročný. Hlavní mód umožňuje bezpečnější autentizaci obou stran než agresivní mód.

2.9.2 IKE fáze 2

• Cílem IKE fáze 2 je vytvoření dvou protisměrných IPsec SA.

• Pomocí Diffie-Hellmanova algoritmu se vytvoří sdílený tajný klíč, kterým se budou symetricky šifrovat přenášená uživatelská data. Je také možné pro tento účel použít klíč vytvořený v IKE fázi 1, kterým se šifruje komunikace v rámci IKE spojení.

• IKE fáze 2 používá jen jeden mód – rychlý mód (quick mode).

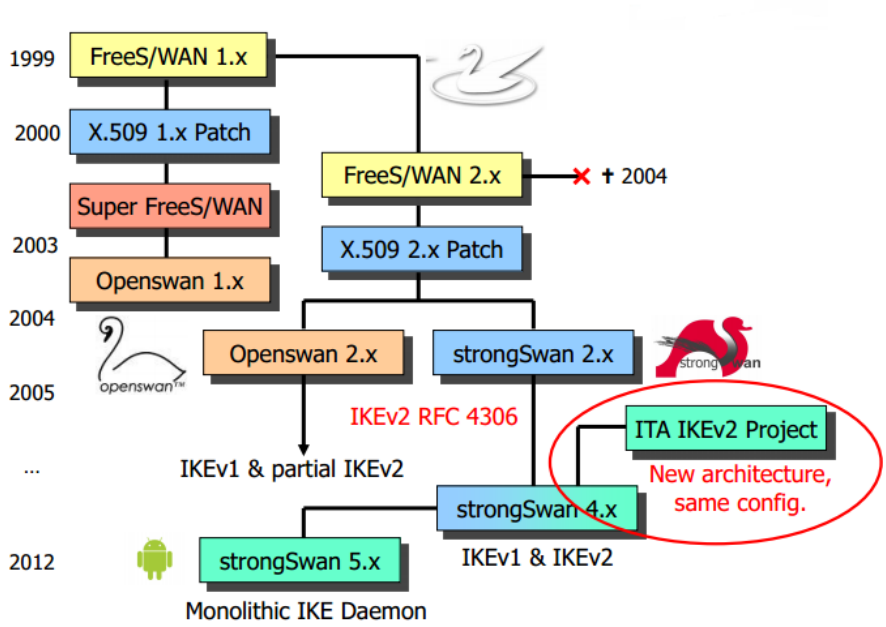
Při zpracování kapitoly jsem čerpal z [2], [5], [7] a [8].

3 strongSwan

StrongSwan je kompletní řešení Open Source VPN pro operační systém Linux. Šířen pod GNU General Public License (GPLv2) zdrojový kód lze volně stáhnout z www.strongswan.org. StrongSwan implementuje protokoly IKEv1 (RFC 2409) a IKEv2 (RFC 4306), které jsou potřebné k nastavení zabezpečeného tunelové spojení IPsec.

3.1 Historie a vývojové etapy

Projekt strongSwan byl zahájen v březnu roku 2004. Vychází z FreeS/WAN projektu (www.freeswan.org), vývoj oficiálně přerušen jeho sponzorem Johnem Gilmorem. Openswan a strongSwan zpočátku sdíleli stejný kód , včetně X.509 patche, který přidal podporu certifikátů a čipových karet u FreeS/WAN. Ale vzhledem k tomu, že Openswan spíše upřednostňoval VPN s podporou IKE v agresivním režimu, strongSwan se zaměřuje na certifikáty a čipové karty používané pro autentizaci. Obrázek 3.1 nám ukazuje celou vývojovou etapu. Poslední verzí je momentálně 5.1.1, která podporuje nejmodernější standardy a také mobilní zařízení.



Obrázek 3.1: Historie a vývojová etapa strongSwan [13]

3.2 Vlastnosti strongSwan VPN

Hlavní vlastnosti softwaru strongSwan jsou následující:

- Běží na linuxových jádrech 2.6 pomocí nativního NETKEY IPsec
- Umožňuje rychlé nastavení připojení VPN prostřednictvím IKEv1 a IKEv2 protokolů (RFC 5996)
- Automaticky vkládá a maže pravidla brány firewall a IPsec politik
- Dynamické přidělování IP adres a aktualizace rozhraní s protokolem MOBIKE (RFC 4555)
- Podporuje šifrování AES (128/192/256 bit), Camellia a 3DES
- Rychlá výměna klíčů pomocí Diffie – Hellmanova algoritmu (skupiny 1,2 a 5)
- Podpora NAT a přiřazení virtuálních IP adres
- IP adresy spravované pomocí IKE démona, nebo SQL databáze
- Má vlastní Peer Detection (RFC 3706), které se stará o vytvořené tunely
- Ověření uživatele na základě certifikátů, nebo předsdílenými klíči
- Získávání a zachytávání neplatných certifikátů přes HTTP nebo LDAP
- Plně podporuje online ověření certifikátů OCSP (RCF 2560)
- Silné zásady IPsec založené na použití certifikační autority
- Použití X.509 certifikátů
- Uložení RSA soukromých klíčů a certifikátů na čipové kartě (PKCS # 11 rozhraní)
- Xauth ověřování v IKEv1 hlavním režimu. IKEv2 – vícenásobné ověřování spojení
- Podpora EAP protokolů. Plná spolupráce IKEv2 s Windows 7 a Windows Server 2008 R2.

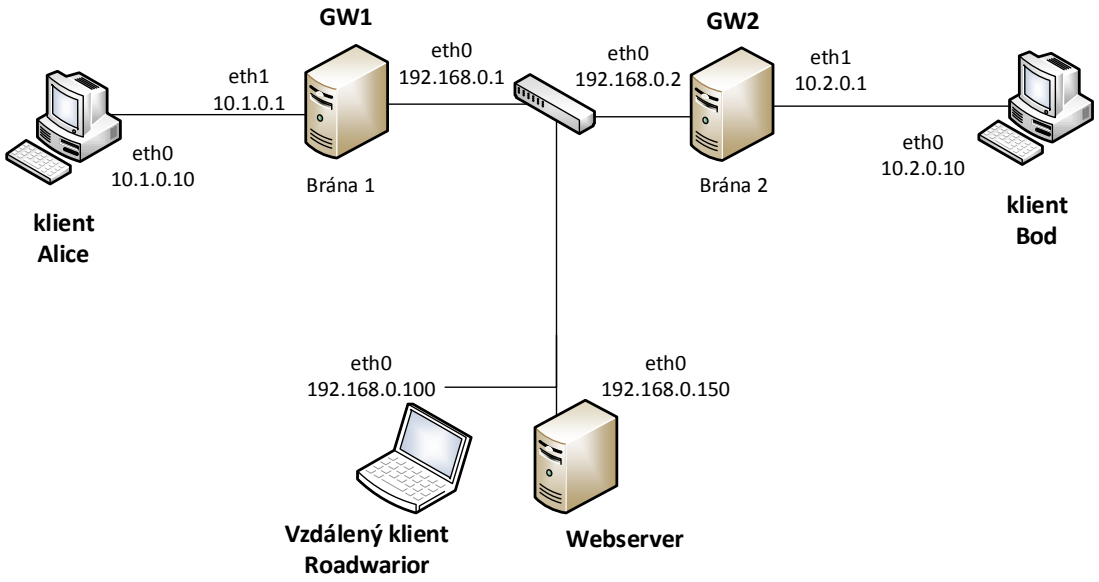
Podporované operační systémy: Linux 2.6 / 3.x, Android 2.x a vyšší, FreeBSD 7.x / 8.x, Mac OS X 10.5 - 10.7

Hardwarové platformy (32/64 bit): Intel, Via, AMD, ARM, MIPS (např. Freescale, Marvell, 16-core Cavium Octeon), PowerPC

Zdrojový kód: 100% napsaný v C, s objektově orientovaným přístupem

3.3 Zabezpečení sítě

strongSwan je kompletní IPsec řešení, které poskytuje šifrování a autentizaci serverů a klientů. To se využívá k zajištění komunikace se vzdálenými sítěmi.



Obrázek 3.2: Zabezpečení sítě

Brána/ Gateway - je obvykle firewall, ale může to být každý host v rámci dané sítě. Pro malou síť slouží jako DNS a DHCP server.

Klienti pro vzdálený přístup / Roadwarrior – obvykle se jedná o notebooky a další mobilní zařízení, připojující se ze vzdáleného místa k dané síti.

Spojení dvou hostů / Host-host : To může být vzdálený webový server nebo záložní systém. To je na obrázku znázorněno hostitelem webserver a některou z bran. Nebo se jedná o spojení dvou bran, v obrázku 3.2 to je GW1 a GW2.

Dálkové spojení / Site-to-Site : Hosti ve dvou nebo více podsítích by měli mít přístup k sobě navzájem. Opět s odkazem na obrázek 4.2, dvě podsítě 10.1.0.0/24 a 10.2.0.0/24 za branami **GW1 a GW2**, se spojí tak, aby spolu **Alice** a **Bob** mohli bezpečně komunikovat.

3.4 Relace IKE_SA

strongSwan je v podstatě IPsec démon, který používá IKE protokoly (IKEv1 a IKEv2), které slouží k navázání bezpečnostních asociací (SA) mezi dvěma zařízeními. IKE poskytuje silnou autentizaci obou uživatelů a vytváří unikátní šifrovací klíče relace. Taková relace IKE je často označován IKE_SA. Dále poskytuje prostředky pro výměnu informací o konfiguraci a vyjednaném IPsec SA. Definuje síťový provoz, který má být šifrován. IPsec provoz není zpracováván strongSwanem, ale prostřednictvím jádra operačního systému (kernelem) a protokolu IPsec.

3.5 Autentizace

Způsoby, které nám strongSwan nabízí pro ověření účastníků:

Public Key Authentication – používá RSA nebo ECDSA X.509 certifikáty. Pro ověření druhého účastníka. Certifikáty mohou být s vlastním podpisem, v takovém případě musí být nainstalovány u všech účastníků, nebo podepsány certifikační autoritou (CA).

Pre-Shared-Key (PSK) – předsdíleným klíčem, jeho implementace je snadná, ale vyžaduje bezpečný klíč. Proto se tato metoda nedoporučuje pro rozsáhlé nasazení.

Extensible Authentication Protocol (EAP) - zahrnuje několik možných metod ověřování, některé z nich jsou založeny na ověřování uživatelského jména a hesla (EAP-MD5, EAP-MSCHAPv2, EAP-GTC), nebo na certifikátu (EAP-TLS)

eXtended Authentication (Xauth) - poskytuje flexibilní rámec pro ověřování v rámci IKEv1. To se hlavně používá pro autentizaci pomocí uživatelského jména a hesla. Používá se obecně jako metoda druhé autentizace společně s PSK.

3.6 Konfigurační soubory

Konfigurační soubory používané strongSwan, jsou následující:

ipsec.conf: umožňuje konfiguraci připojení IPsec

ipsec.secrets: obsahuje klíče (sdílené tajné klíče, soukromé klíče)

ipsec.d: zde ukládáme certifikáty a soukromé klíče

strongswan.conf: konfigurace modulů démona

Konfigurační soubory se nacházejí v kořenovém adresáři systému, ve složce /etc.

terminologie: vlevo (left) a vpravo (right), jak se používá v ipsec.conf souboru, označují dva koncové body.

- vlevo znamená **místní** peer, tedy ten, na kterém je uložena konfigurace
- vpravo je **vzdálený** peer, ke kterému se připojujeme

Mřížka na začátku značí komentáře v konfiguračních souborech.

3.7 Stažení a instalace

a) Instalace pomocí správy balíků:

```
apt-get install strongswan
```

b) Pomocí kompilace zdrojových kódů:

Potřebné balíčky k instalaci:

`gcc` – sada překládačů, která se standardně využívá u open source systémů

`make` – utilita pro automatizaci překladu zdrojových kódů

`libgmp3-dev` – aritmetická knihovna

```
apt-get install gcc
apt-get install make
apt-get install libgmp3-dev
```

Stažení programu (verze 5.1.1)

```
wget http://download.strongswan.org/strongswan-5.1.1.tar.bz2
```

Rozbalení archivu a vstup do rozbalené složky

```
tar xjvf strongswan-5.1.1.tar.bz2; cd strongswan-5.1.1
```

Kompilace zdrojových souborů

```
./configure --prefix=/usr --sysconfdir=/etc
```

Po kompilaci je zobrazen souhrn s nainstalovanými moduly.

```
strongSwan will be built with the following plugins
-------------------------------------------------------
libstrongswan: aes des rc2 sha1 sha2 md5 random nonce x509
revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp
dnskey sshkey pem fips-prf gmp xcbc cmac hmac
libcharon:     socket-default stroke updown xauth-generic
libhydra:      attr kernel-netlink resolve
libtnccs:
```

Instalace

```
make
make install
```

Při zpracování kapitoly jsem čerpal z [3] a [4].

4 Konfigurace VPN sítí s využitím softwaru strongSwan

Aplikace je testována na různých verzích Linuxu - Lubuntu, Ubuntu 13.10 a xUbuntu. Jsou využity poslední stabilní verze – strongSwan 5.1.1 a 5.1.2. PC s nainstalovaným softwarem strongSwan je v některých spojeních využito jako linuxová brána.

Podrobnější popis konfiguračních souborů:

/ etc / ipsec.conf:

- config setup – definuje základní konfigurační parametry daného spojení
- conn <name> - definice daného spojení
- ca <name> - parametry certifikační autority

Dokumentace dle [10].

/ etc / ipsec.secrets - soubor obsahuje neomezený počet typů zabezpečení, certifikátů a klíčů

- RSA definuje soukromý klíč RSA
- ECDSA definuje ECDSA soukromý klíč
- P12 definuje PKCS #12
- PSK definuje předsdílené klíče
- EAP definuje pověření EAP
- NTLM definuje pověření NTLM
- Xauth definuje pověření Xauth
- PIN definuje čipových karet PIN

Dokumentace dle [11].

/etc/ipsec.d/

- **private** - obsahuje RSA a ECDSA soukromé klíče a soubory
- **certs** – obsahuje X.509 a PGP certifikáty
- **crls** – listy odvolaných certifikátů
- **cacerts** – certifikáty certifikační autority
- **ocspcerts** – důvěryhodné OCSP osvědčení
- **aacerts** důvěryhodné certifikáty a povolení
- **acerts** – atributy certifikátů
- **reqs** - PKSC#10 žádosti o certifikáty

/etc/strongswan.conf - základní konfigurační soubor, kde definujeme jednotlivé moduly, např. šifry, hashovací algoritmy a typy certifikátů, které k dané konfiguraci potřebujeme.

Dokumentace dle [12].

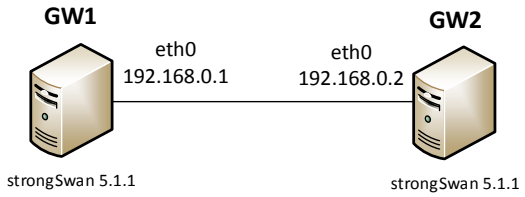
4.1 Konfigurace založené na autentizaci pomocí předsdílených klíčů

Kapitola se zabývá praktickými konfiguracemi spojení typu host-host, site-to-site a vzdáleného klienta, kdy autentizace je založena na předsdíleném klíči. U všech praktických ukázek jsou uvedeny konfigurační soubory, výpisy sestaveného spojení a zachycení zabezpečeného provozu programem Wireshark.

4.1.1 Konfigurace spojení host-host s předsdíleným klíčem (PSK)

Spojení dvou počítačů, na kterých běží software strongSwan. Tato konfigurace představuje tunelovací režim s protokolem ESP (spojení host-host-tunnel), strongSwan je defaultně v tomto režimu takto nastaven. Spojení s názvem host-host-transport je zabezpečení AH protokolem v transportním režimu.

V aktuální verzi strongSwan 5.1.1 se používá IKE démon Charon a nahradil staršího démona Pluta, který byl používán ve verzích 4.x. StrongSwan nepodporuje kombinaci protokolů AH + ESP. Další parametry spojení jsou uvedeny v komentářích konfiguračních souborů.



Obrázek 4.1: Zobrazení schématu spojení host-host

GW1 - konfigurační soubory

```
#ipsec.conf - strongSwan IPsec configuration file
conn %default                  #defaultní parametry pro všechna spojení
      ikelifetime=60m          #doba po které vyprší IKE SA
      keylife=20m              # doba platnosti klíče
      keyingtries=1            # počet pokusů k vyjednání spojení
      keyexchange=ikev2        #verze protokolu ikev2
      authby=secret            #autentizace předsdíleným klíčem

conn host-host-tunnel          #název definovaného spojení
            left=192.168.0.1   #levá strana (zdroj)
            leftid=@gw1        #identifikátor levé strany
            leftfirewall=yes   #povolení pravidel IP tables
            right=192.168.0.2  #pravá strana (cíl)
            rightid=@gw2       #identifikátor pravé strany
            auto=add           #načtení spojení a start ike démona
```

```
conn host-host-transport
          left=192.168.0.1#levá strana (zdroj)
          leftid=@gw1     #identifikátor levé strany
          leftfirewall=yes#povolení pravidel IP tables
          right=192.168.0.2#pravá strana (cíl)
          rightid=@gw2    #identifikátor pravé strany
          type=transport  #transportní režim
          ah=sha1-md5-modp1024! #použité šifry a velikost klíče D-H
algoritmu
          auto=add        #načtení spojení a start ike démona
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
@gw1  @gw2  :  PSK  0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL  #předsdílený
#klíč
```

 # strongswan.conf - strongSwan configuration file

```
charon {
     load = aes des sha1 sha2 md5 gmp random nonce hmac stroke
kernel-netlink socket-default updown }
     libstrongswan {
      dh_exponent_ansi_x9_42 = no
     }#načtení potřebných modulů srongSwan pro sestavení spojení
```

GW2 - konfigurační soubory

Konfigurace je téměř stejná jak na GW1, s rozdílem identifikátorů a IP adres.

#ipsec.conf - strongSwan IPsec configuration file

```
conn %default              #defaultní parametry pro všechna spojení
     ikelifetime=60m       #doba po které vyprší IKE SA
     keylife=20m           # doba platnosti klíče
     keyingtries=1         # počet pokusů k vyjednání spojení
     keyexchange=ikev2     #verze protokolu ike
     authby=secret         #autentizace předsdíleným klíčem
```

```
conn host-host -tunnel      #název definovaného spojení
        left=192.168.0.2#levá strana (zdroj)
        leftid=@gw2     #identifikátor levé strany
        leftfirewall=yes#povolení pravidel IP tables
        right=192.168.0.1#pravá strana (cíl)
        rightid=@gw1    #identifikátor pravé strany
        auto=add        #načtení spojení a start ike démona

conn host-host -transport
        left=192.168.0.2#levá strana (zdroj)
        leftid=@gw2     #identifikátor levé strany
        leftfirewall=yes#povolení pravidel IP tables
        right=192.168.0.1#pravá strana (cíl)
        rightid=@gw1    #identifikátor pravé strany
        type=transport  #transportní režim
        ah=sha1-md5-modp1024! #šifra-hash-DH-algoritmus
        auto=add        #načtení spojení a start ike démona
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
@gw1 @gw2 : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL #před sdílený
#klíč
```

 # strongswan.conf - strongSwan configuration file

```
charon {
     load = aes des sha1 sha2 md5 gmp random nonce hmac stroke
kernel-netlink socket-default updown #načtení potřebných modulů
srongswanu
     }
     libstrongswan {
      dh_exponent_ansi_x9_42 = no
} #načtení všech modulů ike démona charon
```

Postup spouštění nastavených konfigurací:

Po nastavení všech konfiguračních souborů, spustíme na branách (GW1 a GW2) IPsec démona, příkazem `ipsec start`, viz obrázek 4.2.

```
Starting strongSwan 5.1.1 IPsec [starter]...
!! Your strongswan.conf contains manual plugin load options for charon.
!! This is recommended for experts only, see
!! http://wiki.strongswan.org/projects/strongswan/wiki/PluginLoad
charon is already running (/var/run/charon.pid exists) -- skipping daemon start
starter is already running (/var/run/starter.charon.pid exists) -- no fork done
```

Obrázek 4.2: Spuštění strongSwanu příkaz– ipsec start

4.1.2 Sestavení spojení host-host v tunelovacím režimu s protokolem ESP

Poté příkazem `ipsec up <název spojení>`, vytvoříme zabezpečený tunel. V tomto případě: `ipsec up host-host-tunnel.` Na obrázku 4.3 je vidět proces úspěšně sestaveného spojení. Jsou zde vidět zdrojové a cílové adresy, použité porty, vzájemná autentizace, doba životnosti IKE_SA. Pokud v konfiguračních souborech bran `ipsec.conf` nezakážeme protokol MOBIKE parametrem `mobike=no`, strongSwan automaticky komunikuje na portu UDP 4500 a to i v případě, že nebyl zjištěn žádný NAT.

```
initiating IKE_SA host-host-tunnel[1] to 192.168.0.2
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 192.168.0.1[500] to 192.168.0.2[500] (676 bytes)
received packet: from 192.168.0.2[500] to 192.168.0.1[500] (432 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
authentication of 'gw1' (myself) with pre-shared key
establishing CHILD_SA host-host-tunnel
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE
_SUP) N(ADD_4_ADDR) N(EAP_ONLY) ]
sending packet: from 192.168.0.1[4500] to 192.168.0.2[4500] (380 bytes)
received packet: from 192.168.0.2[4500] to 192.168.0.1[4500] (236 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(ADD
_4_ADDR) ]
authentication of 'gw2' with pre-shared key successful
IKE_SA host-host-tunnel[1] established between 192.168.0.1[gw1]...192.168.0.2[gw
2]
scheduling reauthentication in 3252s
maximum IKE_SA lifetime 3432s
connection 'host-host-tunnel' established successfully
```

Obrázek 4.3: Sestavení spojení host-host-tunnel

Příkazem `ipsec statusall` získáme podrobnější výpis o stavu ipsec spojení. U tohoto spojení vidíme jeho název, stav, zdrojovou a cílovou IP adresu. V obrázku 4.4 je žlutým obdélníkem vyznačený režim spojení, jak již bylo uvedeno výše, jedná se o tunelovací režim s ESP protokolem pro zabezpečení přenosu. Z uvedeného výpisu vidíme i druhé spojení `host-host-transport`, které ovšem není aktivní.

```
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.2.0-29-generic, i686):
  uptime: 78 seconds, since Mar 02 19:07:05 2014
  malloc: sbrk 135168, mmap 0, used 75272, free 59896
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled:
 3
  loaded plugins: charon aes des sha1 sha2 md5 gmp random nonce hmac stroke kern
el-netlink socket-default updown
Listening IP addresses:
  10.0.2.15
  192.168.0.1
Connections:
host-host-tunnel:  192.168.0.1...192.168.0.2  IKEv2
host-host-tunnel:   local:  [gw1] uses pre-shared key authentication
host-host-tunnel:   remote: [gw2] uses pre-shared key authentication
host-host-tunnel:   child:  dynamic === dynamic TUNNEL
host-host-transport:   child:  dynamic === dynamic TRANSPORT
Security Associations (1 up, 0 connecting):
host-host-tunnel[1]: ESTABLISHED 42 seconds ago, 192.168.0.1[gw1]...192.168.0.2[
gw2]
host-host-tunnel[1]: IKEv2 SPIs: b4317437d21e0795_i* b0e3203f12e77748_r, pre-sha
red key reauthentication in 52 minutes
host-host-tunnel[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2
048
host-host-tunnel{1}:  INSTALLED, TUNNEL, ESP SPIs: ca95ba75_i c4009db6_o
host-host-tunnel{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying i
n 15 minutes
host-host-tunnel{1}:   192.168.0.1/32 === 192.168.0.2/32
```

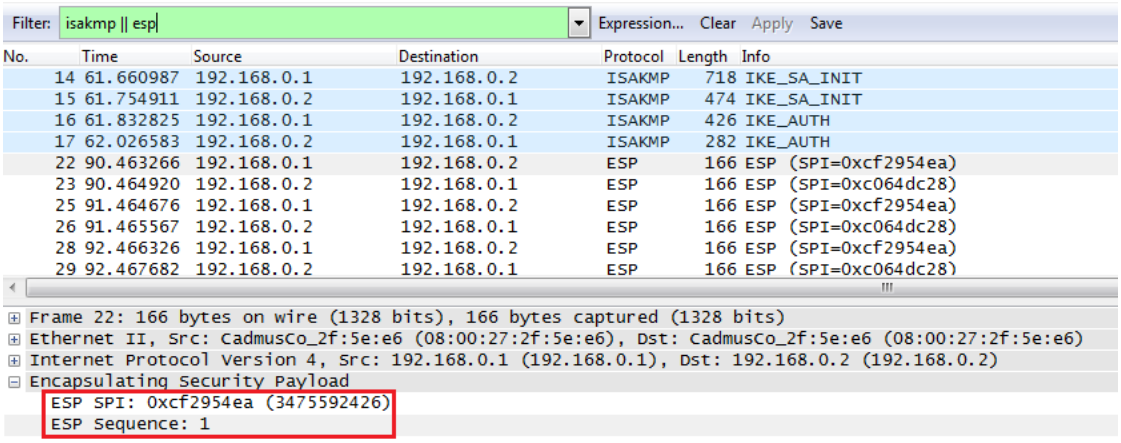
Obrázek 4.4:Tunelovací režim s ESP protokolem – ipsec statusall

Celý proces sestavení je odchycen v programu Wireshark. Zde jsou vidět pakety ISAKMP protokolu, ve kterých jsou zapouzdřeny zprávy protokolu IKE. Jeho strukturu vidíme na obrázku 4.5.

```
⊞ Frame 14: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits)
⊞ Ethernet II, Src: CadmusCo_2f:5e:e6 (08:00:27:2f:5e:e6), Dst: CadmusCo_2f:5e:e6 (08:00:27:2f:5e:e6)
⊞ Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.2 (192.168.0.2)
⊞ User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
⊟ Internet Security Association and Key Management Protocol
    Initiator cookie: 84ef84a02e8e96e3
    Responder cookie: 0000000000000000
    Next payload: Security Association (33)
    Version: 2.0
    Exchange type: IKE_SA_INIT (34)
  ⊞ Flags: 0x08
    Message ID: 0x00000000
    Length: 676
  ⊞ Type Payload: Security Association (33)
  ⊞ Type Payload: Key Exchange (34)
  ⊞ Type Payload: Nonce (40)
  ⊞ Type Payload: Notify (41)
  ⊞ Type Payload: Notify (41)
```

Obrázek 4.5: Struktura protokolu ISAKMP sestaveného spojení host-host-tunnel

Příkazem `ping 192.168.0.2` z brány GW1 na bránu GW2 je vytvořena komunikace a následně zachycena Wiresharkem. Zachycený provoz vidíme na obrázku 4.6, kde je komunikace zabezpečena již dříve uváděným protokolem ESP. V kolonce „Filter“ nastavíme zobrazení jen těchto dvou protokolů (isakmp || esp).



Obrázek 4.6: Zachycení provozu Wiresharkem – spojení host-host-tunnel

4.1.3 Sestavení spojení host-host v transportním režimu s protokolem AH

Příkazem `ipsec up host-host-transport` vytvoříme spojení mezi dvěma hosty, v transportním režimu s protokolem AH. Výpis vytváření spojení vidíme na obrázku 4.7.

```
root@radim-VirtualBox:/home/radim# ipsec up host-host-transport
initiating IKE_SA host-host-transport[1] to 192.168.0.2
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 192.168.0.1[500] to 192.168.0.2[500] (676 bytes)
received packet: from 192.168.0.2[500] to 192.168.0.1[500] (432 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
authentication of 'gw1' (myself) with pre-shared key
establishing CHILD_SA host-host-transport
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE
_SUP) N(ADD_4_ADDR) N(EAP_ONLY) ]
sending packet: from 192.168.0.1[4500] to 192.168.0.2[4500] (380 bytes)
received packet: from 192.168.0.2[4500] to 192.168.0.1[4500] (236 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(ADD
_4_ADDR) ]
authentication of 'gw2' with pre-shared key successful
IKE_SA host-host-transport[1] established between 192.168.0.1[gw1]...192.168.0.2
[gw2]
scheduling reauthentication in 3303s
maximum IKE_SA lifetime 3483s
connection 'host-host-transport' established successfully
```

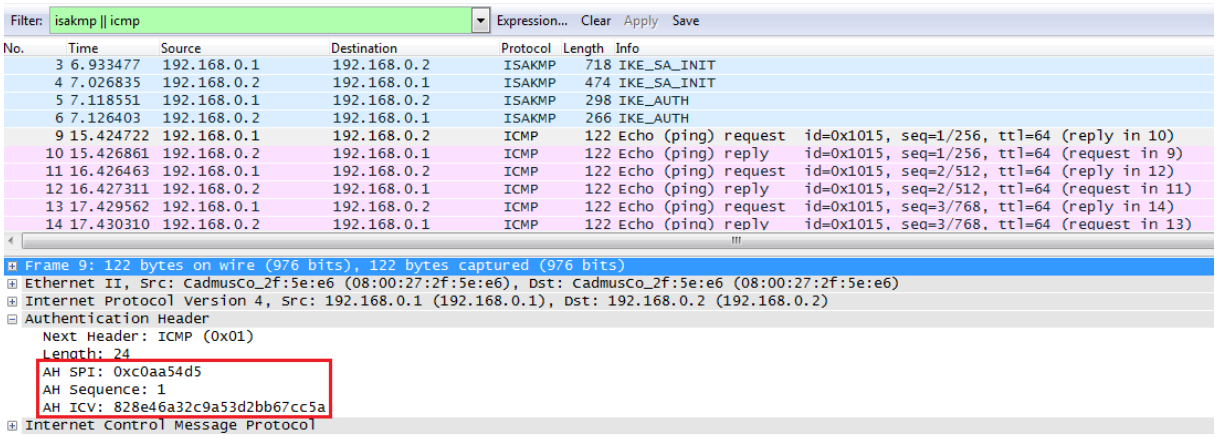
Obrázek 4.7: Sestavení spojení host-host-transport

Podrobný výpis sestaveného spojení získáme opět příkazem `ipsec statusall`, ve kterém vidíme parametry vytvořeného přenosu. Žlutě je označen režim a protokol (obrázek 4.8).

```
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.2.0-29-generic, i686):
  uptime: 4 minutes, since Mar 02 23:10:34 2014
  malloc: sbrk 135168, mmap 0, used 73976, free 61192
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled:
 3
  loaded plugins: charon aes des sha1 sha2 md5 gmp random nonce hmac stroke kern
el-netlink socket-default updown
Listening IP addresses:
  10.0.2.15
  192.168.0.1
Connections:
host-host-transport:  192.168.0.1...192.168.0.2  IKEv2
host-host-transport:   local:  [gw1] uses pre-shared key authentication
host-host-transport:   remote: [gw2] uses pre-shared key authentication
host-host-transport:   child:  dynamic === dynamic TRANSPORT
Security Associations (1 up, 0 connecting):
host-host-transport[1]: ESTABLISHED 3 minutes ago, 192.168.0.1[gw1]...192.168.0.
2[gw2]
host-host-transport[1]: IKEv2 SPIs: 9f03bb723019c397_i* ad94daab06f3d273_r, pre-
shared key reauthentication in 48 minutes
host-host-transport[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MOD
P_2048
host-host-transport{1}:  INSTALLED, TRANSPORT, AH SPIs: c37e64e8_i cf01ff35_o
host-host-transport{1}:  HMAC_SHA1_96, 256 bytes_i (4 pkts, 214s ago), 256 bytes
_o (4 pkts, 214s ago), rekeying in 11 minutes
host-host-transport{1}:   192.168.0.1/32 === 192.168.0.2/32
```

Obrázek 4.8: Transportní režim s AH protokolem – ipsec statusall

Obrázek 4.9 zobrazuje zachycený provoz programem wireshark. Jedná se o spojení host-host-transport a provoz byl vytvořen příkazem ping z adresy 192.168.0.1 na adresu 192.168.0.2. V obrázku 4.9 je červeným rámečkem označen protokol AH, který je přidán za původní IP záhlaví. Je zde uvedena hodnota SPI (Security parameter index) a číslo sekvence.



Obrázek 4.9: Zachycení provozu Wiresharkem – spojení host-host-transport

4.1.4 Vzdálený přístup s využitím předsdíleného klíče pro autentizaci

Tato implementace se zaměřuje na přístup vzdáleného klienta na bránu GW. Autentizace je založena na předsdílených klíčích. Po úspěšném vytvoření tunelu IPsec, nám příkaz `leftfirewall=yes` umožní povolit tunelový provoz. Sestavené spojení otestujeme příkazem ping Vzdáleného klienta na klienta Alice, který je za branou GW.

Nastavení síťových rozhraní a pravidel iptables:

Zapnutí směrování neboli předávání paketů z jednoho síťového rozhraní na druhé:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

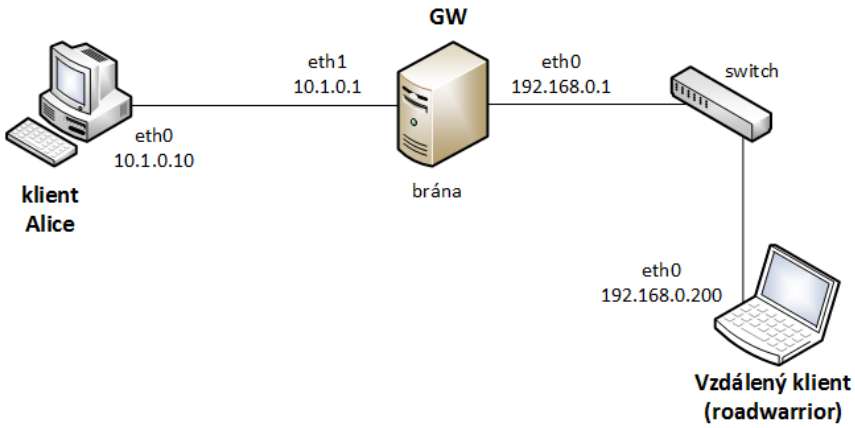
Provedené nastavení zkontrolujeme následujícím příkazem:

```
cat /proc/sys/net/ipv4/ip_forward
```

Pokud je routování zapnuto vypíše 1, pokud je vypnuto vypíše 0.

Nastavení pravidel IPTABLES na bráně GW.

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
```



Obrázek 4.10: Schéma vzdálený přístup

GW – konfigurační soubory

`#/etc/ipsec.conf - strongSwan IPsec configuration file`

```
config setup
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev2
```

```
      authby=secret

conn home
      left=192.168.0.1
      leftsubnet=10.1.0.0/24
      leftfirewall=yes
      right=%any
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
      192.168.0.200 : PSK 0sjVzONCF02ncsgiSlmIXeqhGN
```

 # /etc/strongswan.conf - strongSwan configuration file

```
      charon {
load = aes des sha1 sha2 md5 gmp random nonce hmac stroke kernel-
netlink socket-default updown
}
```

Vzdálený klient – konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev2
      authby=secret

conn home
      left=192.168.0.200
      leftfirewall=yes
      right=192.168.0.1
      rightsubnet=10.1.0.0/24
      auto=add
```

```
#/etc/ipsec.secrets - strongSwan IPsec secrets file

      192.168.0.200  : PSK 0sjVzONCF02ncsgiSlmIXeqhGN

#/etc/strongswan.conf - strongSwan configuration file

      charon {
load = aes des sha1 sha2 md5 gmp random nonce hmac stroke kernel-
netlink socket-default updown
}
```

Spojení sestavíme příkazem `ipsec up home` na Vzdáleném klientovi. Na obrázku 4.11 je znázorněn postup vytvoření spojení mezi hosty.

```
initiating IKE_SA home[1] to 192.168.0.1
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 192.168.0.200[500] to 192.168.0.1[500] (676 bytes)
received packet: from 192.168.0.1[500] to 192.168.0.200[500] (440 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(MULT_AUTH)
]
authentication of '192.168.0.200' (myself) with pre-shared key
establishing CHILD_SA home
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE
_SUP) N(ADD_4_ADDR) N(MULT_AUTH) N(EAP_ONLY) ]
sending packet: from 192.168.0.200[4500] to 192.168.0.1[4500] (396 bytes)
received packet: from 192.168.0.1[4500] to 192.168.0.200[4500] (252 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(ADD
_4_ADDR) N(ADD_4_ADDR) ]
authentication of '192.168.0.1' with pre-shared key successful
IKE_SA home[1] established between 192.168.0.200[192.168.0.200]...192.168.0.1[19
2.168.0.1]
scheduling reauthentication in 3327s
maximum IKE_SA lifetime 3507s
connection 'home' established successfully
```

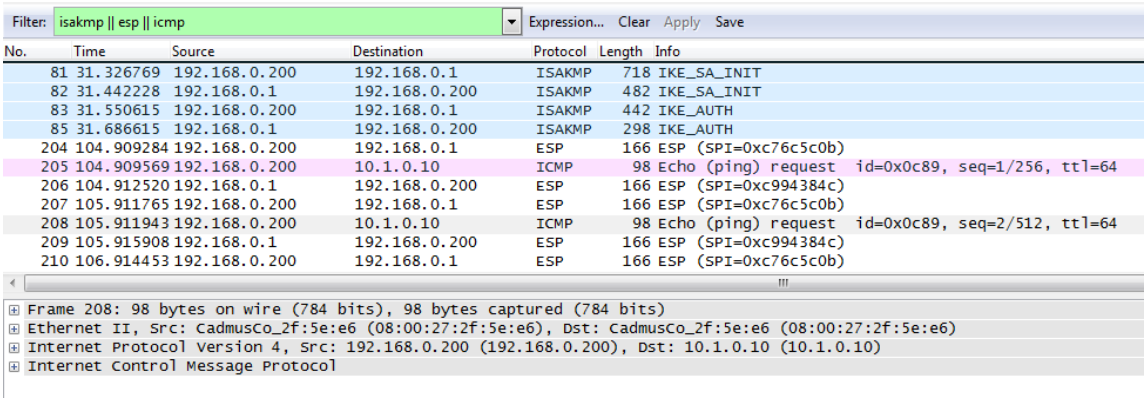
Obrázek 4.11: Vzdálený přístup sestavení spojení

Obrázek 4.12 znázorňuje výpis příkazem `ipsec statusall`, který ukazuje podrobně parametry sestaveného spojení.

```
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.2.0-29-generic, i686):
  uptime: 116 minutes, since Mar 04 12:33:54 2014
  malloc: sbrk 135168, mmap 0, used 76272, free 58896
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled:
 3
  loaded plugins: charon aes des sha1 sha2 md5 gmp random nonce hmac stroke kern
el-netlink socket-default updown
Listening IP addresses:
  10.0.2.15
  192.168.0.200
Connections:
        home:  192.168.0.200...192.168.0.1  IKEv2
        home:   local:  [192.168.0.200] uses pre-shared key authentication
        home:   remote: [192.168.0.1] uses pre-shared key authentication
        home:   child:  dynamic === 10.1.0.0/24 TUNNEL
Security Associations (1 up, 0 connecting):
        home[3]: ESTABLISHED 9 minutes ago, 192.168.0.200[192.168.0.200]...192.1
68.0.1[192.168.0.1]
        home[3]: IKEv2 SPIs: 391a886d0899efce_i* f6bb1298c4a68d9d_r, pre-shared
key reauthentication in 43 minutes
        home[3]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
        home{1}:  INSTALLED, TUNNEL, ESP SPIs: c03e1cac_i ca250028_o
        home{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 5
minutes
        home{1}:   192.168.0.200/32 === 10.1.0.0/24
```

Obrázek 4.12: Vzdálený přístup sestavení spojení

Zachycení provozu programem Wireshark na bráně GW. Proveden ping ze vzdáleného klienta na klienta ve vnitřní sítí Alice. Z výpisu na obrázku 4.13 je vidět zabezpečené tunelové připojení mezi Vzdáleným klientem a bránou GW.



Obrázek 4.13: Zachycení provozu Wiresharkem – vzdálený přístup

4.1.5 Konfigurace site-to-site

V této konfiguraci jsou dvě nezávislé podsítě (10.1.0.0 a 10.2.0.0), které jsou za branami GW1 a GW2, na kterých běží strongSwan a zprostředkovává VPN tunel mezi oběma branami resp. sítěmi. Pro autentizaci je použito předsdíleného klíče, který vidíme v souboru `ipsec.secrets`. IPsec se na jednotlivých branách aktivuje příkazem `ipsec start`, čím spustíme strongSwan, resp. démona Charon. Před spuštěním IPsec démona jsou nastavena následující pravidla síťovým rozhraním:

Zapnutí směrování neboli předávání paketů z jednoho síťového rozhraní na druhé:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

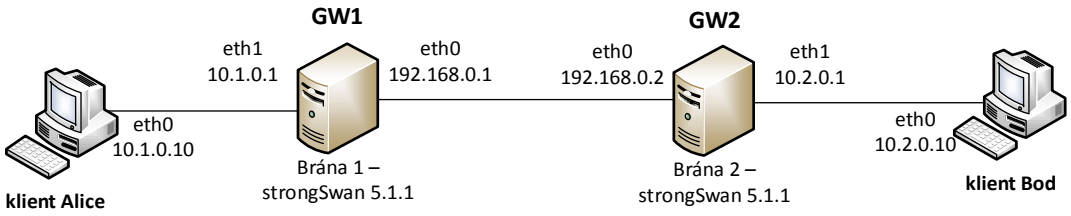
Provedené nastavení zkontrolujeme tímto příkazem:

```
cat /proc/sys/net/ipv4/ip_forward
```

Pokud je směrování zapnuto vypíše 1, pokud je vypnuto vypíše 0.

Nastavení pravidel IPTABLES na obou branách GW1 a GW2 pro přeposílání paketů z jednoho rozhraní na druhé:

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
```



Obrázek 4.14: Schéma zapojení konfigurace site-to-site

GW1 – konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      authby=secret
      keyexchange=ikev2
      mobike=no
conn site-to-site     #název spojení#
      left=192.168.0.1
      leftsubnet=10.1.0.0/24
      leftid=@gw1     #identifikátor levé strany
      leftfirewall=yes
      right=192.168.0.2
      rightsubnet=10.2.0.0/16
      rightid=@gw2    #identifikátor pravé strany
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
@gw1 @gw2 : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL    #předsdílený klíč
```

#/etc/strongswan.conf - strongSwan configuration file

```
      charon {
 load = aes des sha1 sha2 md5 gmp random nonce hmac stroke kernel-netlink socket-default updown
}
```

GW2 – konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      authby=secret
      keyexchange=ikev2
      mobike=no
conn site-to-site     #název spojení
      left=192.168.0.2
      leftsubnet=10.2.0.0/24
      leftid=@gw2
      leftfirewall=yes
      right=192.168.0.1
      rightsubnet=10.1.0.0/24
      rightid=@gw1
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
@gw1 @gw2 : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL
```

#/etc/strongswan.conf - strongSwan configuration file

```
charon {
 load = aes des sha1 sha2 md5 gmp random nonce hmac stroke kernel-
netlink socket-default updown
}
```

Výpis při vytváření spojení je uveden na obrázku 4.15. V této konfiguraci máme neaktivní protokol MOBIKE, tudíž komunikace probíhá jen na portu 500.

```
initiating IKE_SA site-to-site[1] to 192.168.0.2
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 192.168.0.1[500] to 192.168.0.2[500] (676 bytes)
received packet: from 192.168.0.2[500] to 192.168.0.1[500] (432 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
authentication of 'gw1' (myself) with pre-shared key
establishing CHILD_SA site-to-site
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(EAP_O
LY) ]
sending packet: from 192.168.0.1[500] to 192.168.0.2[500] (364 bytes)
received packet: from 192.168.0.2[500] to 192.168.0.1[500] (204 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) ]
authentication of 'gw2' with pre-shared key successful
IKE_SA site-to-site[1] established between 192.168.0.1[gw1]...192.168.0.2[gw2]
scheduling reauthentication in 3263s
maximum IKE_SA lifetime 3443s
connection 'site-to-site' established successfully
```

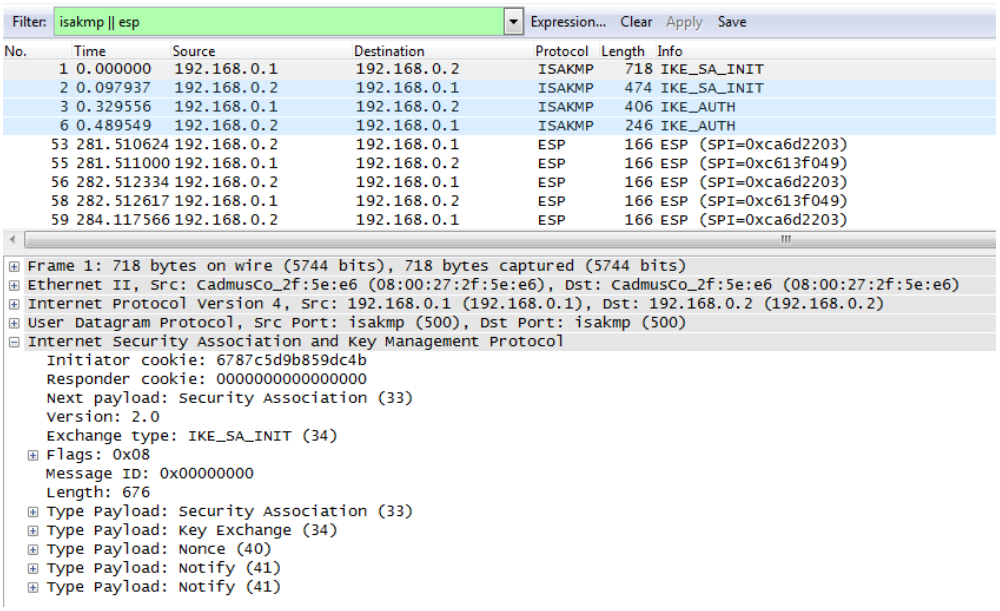
Obrázek 4.15: Sestavení tunelového spojení site-to-site

Výpis podrobností o sestaveném spojení `site-to-site` je provedeno opět příkazem `ipsec statusall` (obrázek 4.16).

```
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.2.0-29-generic, i686):
  uptime: 65 seconds, since Mar 21 16:10:43 2014
  malloc: sbrk 135168, mmap 0, used 74792, free 60376
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled:
 3
  loaded plugins: charon aes des sha1 sha2 md5 gmp random nonce hmac stroke kern
el-netlink socket-default updown
Listening IP addresses:
  10.0.2.15
  10.1.0.1
  192.168.0.1
Connections:
site-to-site:  192.168.0.1...192.168.0.2  IKEv2
site-to-site:   local:  [gw1] uses pre-shared key authentication
site-to-site:   remote: [gw2] uses pre-shared key authentication
site-to-site:   child:  10.1.0.0/24 === 10.2.0.0/24 TUNNEL
Security Associations (1 up, 0 connecting):
site-to-site[1]: ESTABLISHED 63 seconds ago, 192.168.0.1[gw1]...192.168.0.2[gw2]
site-to-site[1]: IKEv2 SPIs: ab6047864d52c6b1_i* b5a4548e6abf1fce_r, pre-shared
key reauthentication in 50 minutes
site-to-site[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
site-to-site{1}:  INSTALLED, TUNNEL, ESP SPIs: c09f80a2_i c1472ca3_o
site-to-site{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 13
 minutes
site-to-site{1}:   10.1.0.0/24 === 10.2.0.0/24
```

Obrázek 4.16: Výpis konfigurace site-site příkazem – ipsec statusall

Zachycený provoz programem Wireshark, ukazuje obrázek 4.17. Opět zobrazen výpis protokolů ISAKMP a ESP



Obrázek 4.17: Zachycení provozu programem Wireshark - spojení site-to-site

4.2 Konfigurace založené na autentizaci pomocí certifikátů certifikační autority

Tato kapitola se věnuje práci s programem XCA, ve kterém je realizována tvorba certifikačních autorit, certifikátů vydaných a podepsaných touto autoritou. Tyto postupy jsou nezbytné pro řešené konfigurace v této kapitole. Takto vytvořené certifikáty jsou následně využity pro otestování konfigurací u spojení typu host-host a vzdáleného klienta.

4.2.1 Popis a práce s programem XCA

Program slouží pro generování a správu soukromých klíčů, certifikátů, žádostí o certifikáty a v neposlední řadě seznamů zneplatněných certifikátů (CRL). Je zdarma a podporuje operační systémy Linux, Windows i Mac OS. Jeho jádro tvoří kryptografické knihovny OpenSSL. Aplikace se ovládá pomocí uživatelského grafického rozhraní. Všechna generovaná data se ukládají do databázového souboru, který je chráněn heslem. Program je dostupný pouze v anglickém jazyce.

Uvedený program je použit pro vytvoření certifikační autority, veřejných klíčů a certifikátů pro účely testování řešených konfigurací.

- **Instalace pod operačním systémem Linux**

  Aplikaci nainstalujeme přes terminál zadáním příkazu:

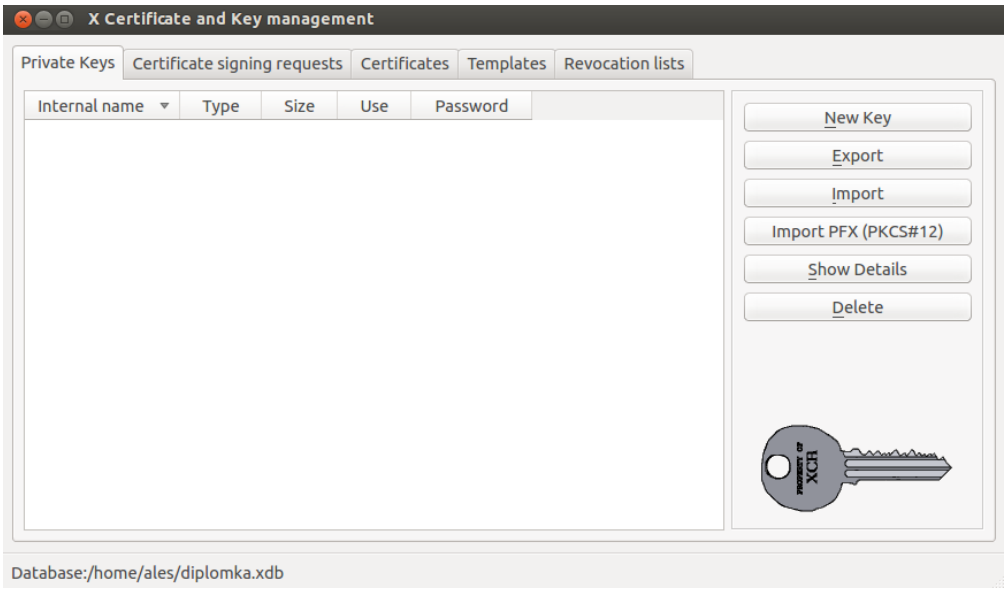
  ```
  apt-get install xca
  ```

- **Instalace pod operačním systémem Windows**

Program je i ve verzi pro tento OS, jako .exe soubor. Dostupný např. zde: http://sourceforge.net/projects/xca/.

- **Představení prostředí programu**

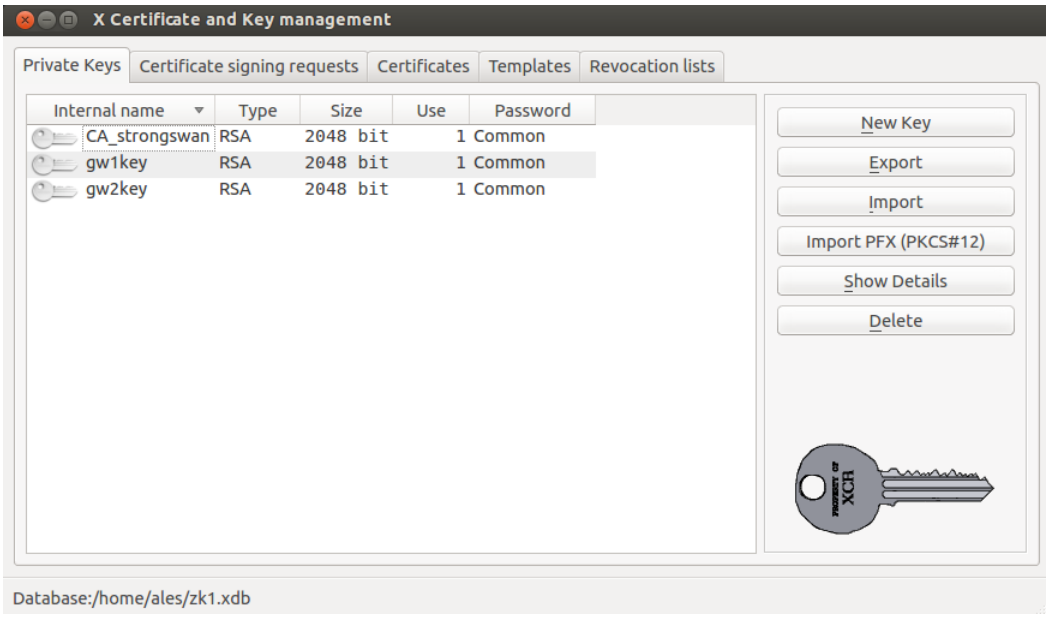
Po spuštění je nejprve vytvořena databáze, kterou chrání heslo. Po vytvoření vlastní databáze, vidíme na obrázku 4.18 hlavní okno programu.



Obrázek 4.18: Program XCA - Hlavní okno

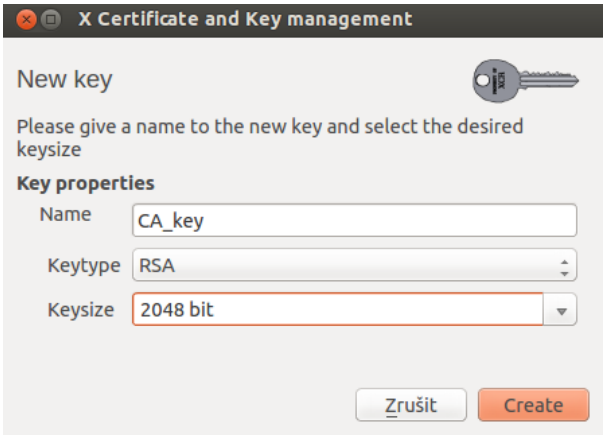
- **Generování soukromých klíčů**

V záložce „Private Keys“ (obrázek 4.19) jsou vytvářeny soukromé klíče, které poté slouží pro podepisování a vydávání certifikátů.



Obrázek 4.19: XCA - Soukromé klíče

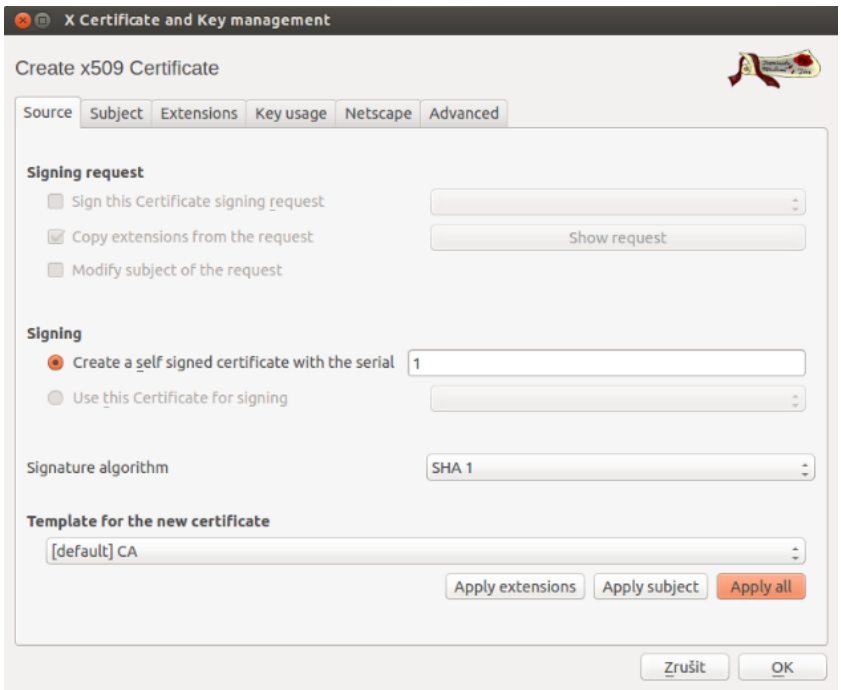
Obrázek 4.20 znázorňuje vytvoření soukromého klíče, jeho jména, typu a velikosti.



Obrázek 4.20: Nový soukromý klíč

- **Vytvoření vlastní certifikační autority**

V hlavním okně (obrázek 4.19) se klikne na záložku „Certifikates“ a v záložce „Source“ (obrázek 4.21) vybereme „Template for the new certifikate“ jako [default] CA – pro vytvoření vlastní certifikační autority.

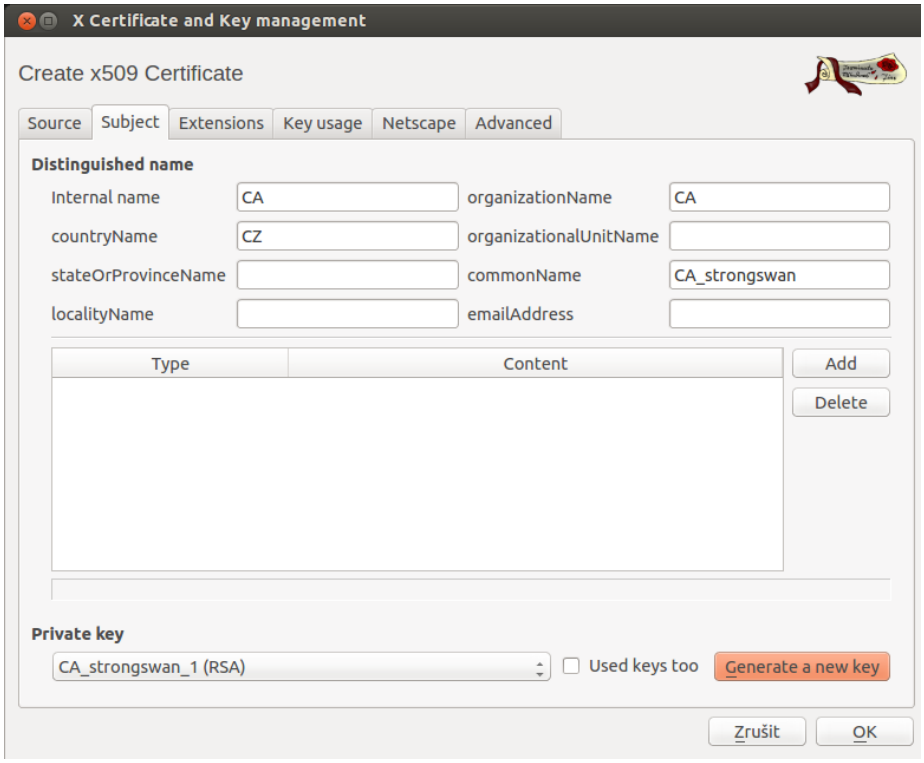


Obrázek 4.21: Certifikát CA

V záložce „Subject“ je vyplněno tzv. dname certifikátu. Minimum pro vytvoření je zobrazeno na obrázku 4.22. Do pole „Internal name“ zadáno interní název certifikátu, dále:

- „countryName“ - zadána hodnotu CZ,
- „commonName“ – zde uvedený název certifikační autority
- „organizationName“ a „organizationalUnitName“ – zde lze volitelně zadat název organizace a organizační složky, které bude provoz autority zajišťovat

Ve spodní části obrazovky vybrán vygenerovaný soukromý klíč (CAkey), pro naši certifikační autoritu.

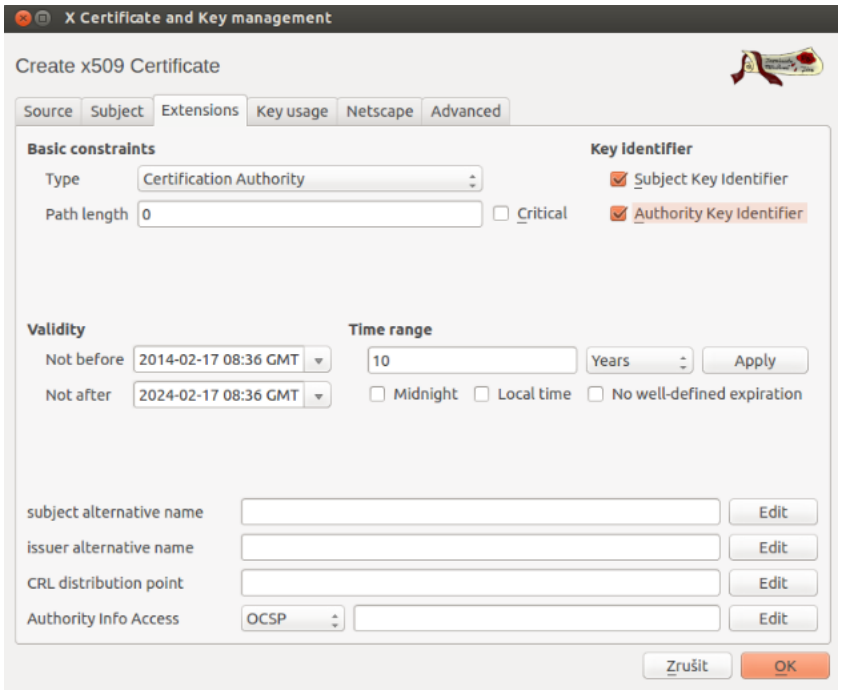


Obrázek 4.22: Certifikační autorita CA_strongswan – identifikační údaje

Záložka „Extensions“ specifikuje rozšíření certifikátu. Pro certifikáty certifikační autority je typické následující nastavení (obrázek 4.23).

V horní části „Basic constraints“ se nastaví pole „Type“ na hodnotu „Certification Authority“. Do pole „Path Length“ se vloží číslo 0, které znamená, že pod touto certifikační autoritou nemohou vzniknout žádné další podřízené certifikační autority. Vpravo nahoře se zaškrtnou obě políčka „Subject Key Identifier“ a „Authority Key Identifier“. Ta do certifikátu přidají identifikátor veřejného klíče vlastníka certifikátu a identifikátor veřejného klíče autority. Tyto identifikátory se používají k sestavení certifikační cesty od certifikátu k certifikátu vydávající CA.

Uprostřed záložky se nastaví doba platnosti certifikační autority a potvrdí tlačítkem „Apply“.



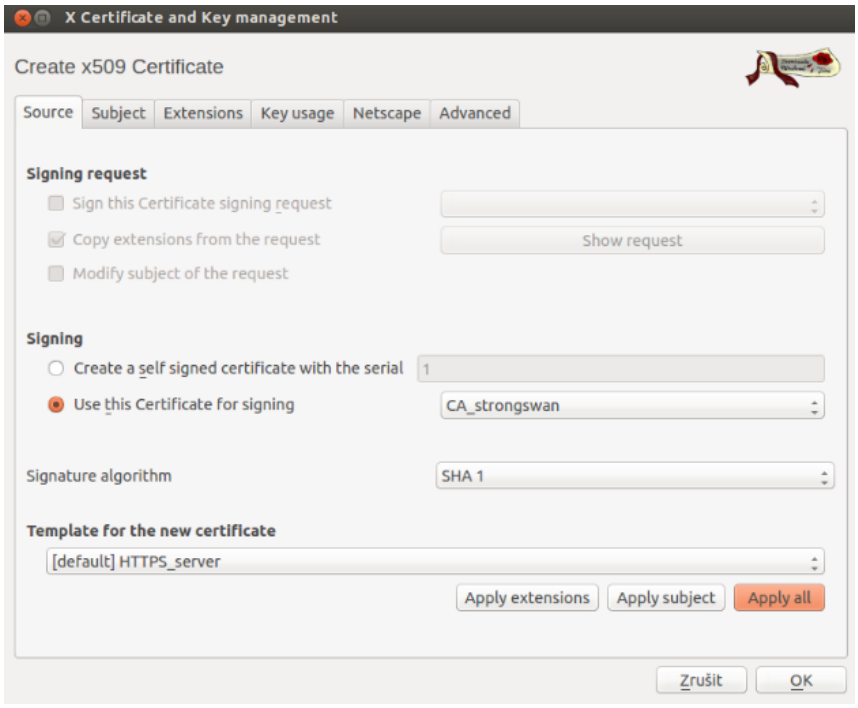
Obrázek 4.23: Rozšíření a další parametry vytvořené CA

Další záložky jsou ponechány v defaultním nastavením. Po provedení všech nastavení naší certifikační autority, potvrzeno stiskem tlačítka OK. Tímto byla vytvořena certifikační autorita, která bude sloužit pro podepisování a vydávání digitálních certifikátů.

- **Vydávání certifikátů vytvořenou certifikační autoritou**

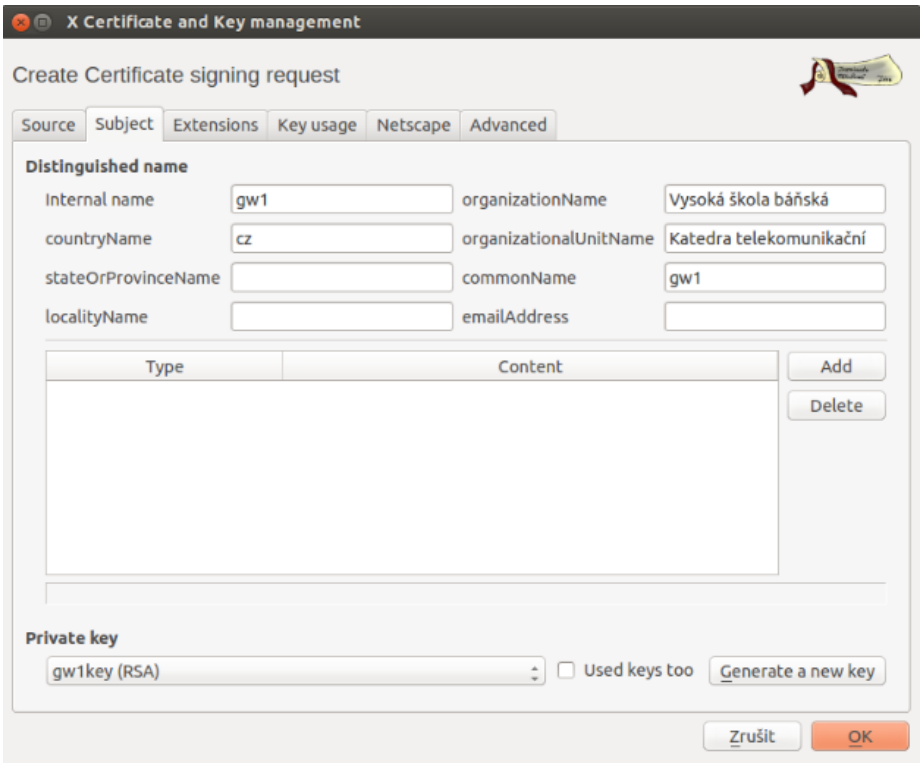
Vytvoření uživatelského certifikátu podepsaného naší certifikační autoritou je provedeno na základě importování žádosti o tento certifikát, nebo vytvořením nového certifikátu. V tomto případě je vytvořen nový certifikát podepsaný certifikační autoritou.

V záložce „Certifikates“ se klikne na „New Certificate“. V „Source“ je nastaveno v kolonce „Signing“ „Use this Certificate for signing“ a vybrán certifikát certifikační autority, kterým je podepsán vydaný certifikát. V menu „Template for the new certifikate“ je vybrána šablona „http_server“, vše potvrzeno tlačítkem „Apply all“.



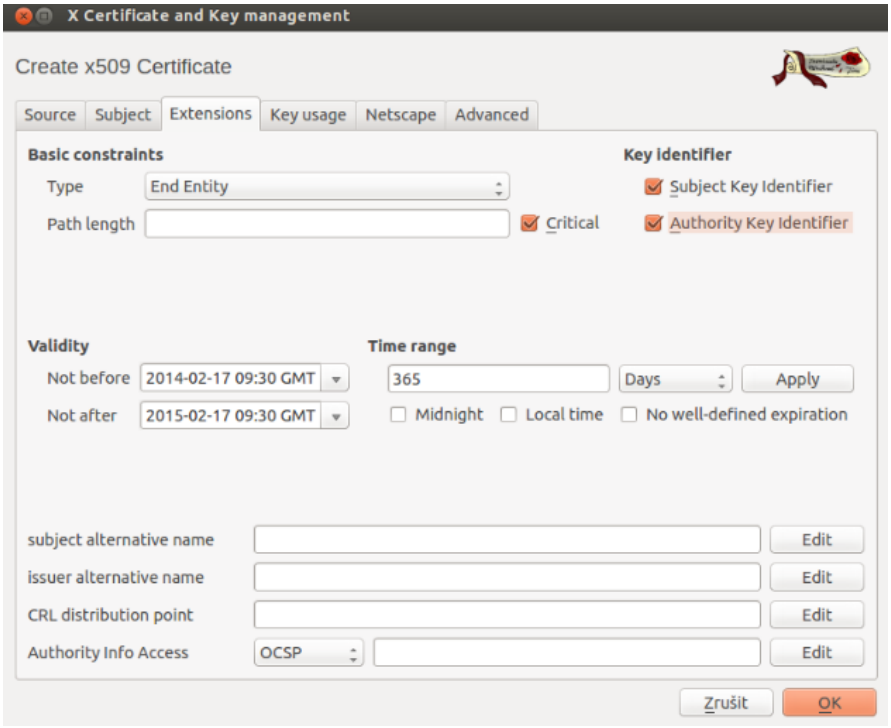
Obrázek 4.24: Certifikát podepsaný CA_strongswan

V záložce „Subject“ je nastaveno tzv. „dname“ certifikátu, jak je uvedeno na obrázku 4.25. Dole vybrán „Private key“, který byl vygenerován v úvodu pro daný certifikát.



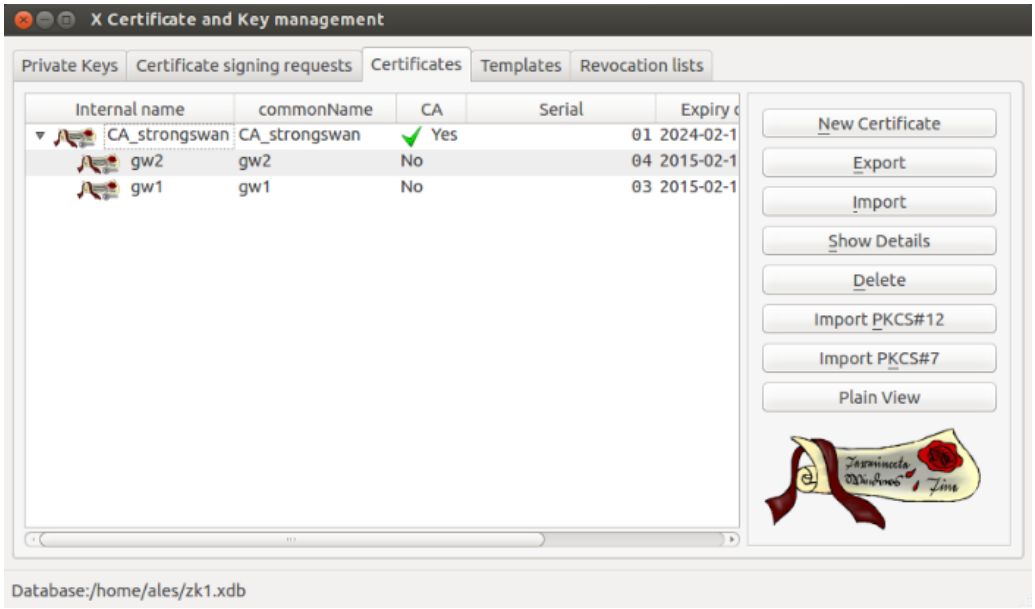
Obrázek 4.25: Vytvoření certifikátu pro bránu GW

V záložce „Extensions“ je vybrán v poli „Type“ typ entity jako - „End Entity“. Zatrhnuty obě zatržítka v oblasti „Key Identifier“, jak vidíme na obrázku 4.26. Uprostřed nastavena platnost vydávaného certifikátu a potvrzeno Apply a OK.



Obrázek 4.26: Certifikát gw1 – nastavení dalších parametrů a rozšíření

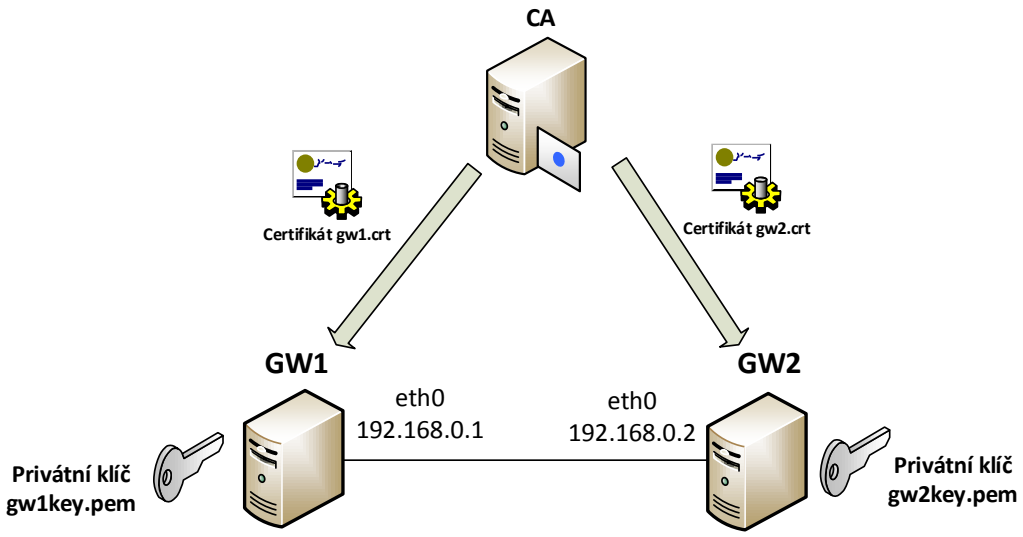
Na obrázku 4.27 je znázorněna vytvořená struktura s certifikační autoritou a dvěma vydanými certifikáty, které jsou podepsané touto autoritou. Nyní se již může každý z certifikátů exportovat do požadovaného formátu a distribuovat na koncová místa v patřičném formátu. Po kliknutí pravým tlačítkem na certifikát jej pomocí volby „Export“ exportujeme do souboru.



Obrázek 4.27: Vytvořené certifikáty pod CA_strongswan

4.2.2 Spojení host-host-cert, použití certifikátů CA a privátních klíčů

Po vytvoření certifikační autority a vydání certifikátů pro obě brány, je nakonfigurováno spojení mezi těmito branami. Rozšíření a umístění certifikátů je rozebráno v tabulce 4.1. Autentizace je provedena na základě těchto certifikátů. Názorné schéma je zobrazeno na obrázku 4.28. Název našeho spojení



Obrázek 4.28: Schéma spojení host-host-cert za použití certifikátů

Tabulka 4.1: Tabulka použitých certifikátů a privátních klíčů u spojení host-host-cert

CA_strongswan – certifikát certifikační autority – umístění /etc/ipsec.d/cacerts	
GW1	**GW2**
Privátní klíč (gw1key.pem) - etc/ipsec.d/private	Privátní klíč (gw2key.pem) - etc/ipsec.d/private
Certifikát gw1.crt – podepsaný soukromým klíčem CA - etc/ipsec.d/certs	Certifikát gw1.crt - podepsaný soukromým klíčem CA - etc/ipsec.d/certs

GW1 – konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
```

```
      keyingtries=1
      keyexchange=ikev2
conn host-host-cert
      left=192.168.0.1
      leftcert=gw1.crt      #certifikát hosta gw1
      leftid="C=cz,   O=gw1,   OU=gw1,   CN=gw1"#identifikátor   dle
certifikátu hosta gw1
      leftfirewall=yes
      right=192.168.0.2
      rightid="C=cz,  O=gw2,  OU=gw2,  CN=gw2"  "#identifikátor  dle
certifikátu hosta gw2
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
      : RSA gw1key.pem"#soukromý klíč hosta gw1
```

#/etc/strongswan.conf - strongSwan configuration file

```
charon {
       load = curl aes des sha1 sha2 md5 pem pkcs1 gmp random nonce
x509 revocation hmac xcbc stroke kernel-netlink socket-default
updown
      } #načtení modulů démona strongSwan
```

GW2 – konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev2
conn host-host-cert
      left=192.168.0.2
      leftcert=gw2.crt      #certifikát hosta gw2
      leftid="C=cz,  O=gw2,  OU=gw2,  CN=gw2"  "#identifikátor  dle
certifikátu hosta gw2
```

```
      leftfirewall=yes
      right=192.168.0.1
      rightid="C=cz,  O=gw1,  OU=gw1,  CN=gw1"  "#identifikátor  dle
certifikátu hosta gw1

      auto=add
#/etc/ipsec.secrets - strongSwan IPsec secrets file

      : RSA gw2key.pem      #soukromý klíč hosta gw2

#/etc/strongswan.conf - strongSwan configuration file
      charon {
       load = curl aes des sha1 sha2 md5 pem pkcs1 gmp random nonce
x509 revocation hmac xcbc stroke kernel-netlink socket-default
updown
      }     #načtení modulů IKE démona strongSwan
```

Po spuštění IKE démona strongSwan příkazem `ipsec start`, je vytvořeno spojení mezi oběma účastníky příkazem `ipsec up host-host-cert` (`host-host-cert` název definovaného spojení v konfiguračním souboru ipsec.conf). Postup sestavení spojení je znázorněn na obrázku 4.29.

```
initiating IKE_SA host[1] to 192.168.0.2
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 192.168.0.1[500] to 192.168.0.2[500] (692 bytes)
received packet: from 192.168.0.2[500] to 192.168.0.1[500] (465 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(MUL
T_AUTH) ]
received cert request for "C=cz, O=ca, OU=ca, CN=CA_strongswan"
sending cert request for "C=cz, O=ca, OU=ca, CN=CA_strongswan"
authentication of 'C=cz, O=gw1, OU=gw1, CN=gw1' (myself) with RSA signature succ
essful
sending end entity cert "C=cz, O=gw1, OU=gw1, CN=gw1"
establishing CHILD_SA host
generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi
 TSr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(MULT_AUTH) N(EAP_ONLY) ]
sending packet: from 192.168.0.1[4500] to 192.168.0.2[4500] (1756 bytes)
received packet: from 192.168.0.2[4500] to 192.168.0.1[4500] (1500 bytes)
parsed IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP)
N(ADD_4_ADDR) ]
received end entity cert "C=cz, O=gw2, OU=gw2, CN=gw2"
  using certificate "C=cz, O=gw2, OU=gw2, CN=gw2"
  using trusted ca certificate "C=cz, O=ca, OU=ca, CN=CA_strongswan"
checking certificate status of "C=cz, O=gw2, OU=gw2, CN=gw2"
certificate status is not available
  reached self-signed root ca with a path length of 0
authentication of 'C=cz, O=gw2, OU=gw2, CN=gw2' with RSA signature successful
IKE_SA host[1] established between 192.168.0.1[C=cz, O=gw1, OU=gw1, CN=gw1]...19
2.168.0.2[C=cz, O=gw2, OU=gw2, CN=gw2]
scheduling reauthentication in 3367s
maximum IKE_SA lifetime 3547s
CHILD_SA host{1} established with SPIs cc700665_i c1b8f6c4_o and TS 192.168.0.1/
32 === 192.168.0.2/32
received AUTH_LIFETIME of 3254s, scheduling reauthentication in 3074s
connection 'host' established successfully
```

Obrázek 4.29: Výpis sestavování spojení host-to-host-cert

Příkazem `ipsec statusall` je zobrazen podrobný výpis sestaveného spojení, který prezentuje obrázek 4.30.

```
root@ubuntu:/home/radim# ipsec statusall
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.11.0-15-generic, x86_64):
  uptime: 2 minutes, since Mar 08 13:37:51 2014
  malloc: sbrk 2297856, mmap 0, used 228000, free 2069856
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
  loaded plugins: charon aes des sha1 sha2 md5 pem pkcs1 gmp random nonce x509 revocation hmac xcbc s
troke kernel-netlink socket-default updown
Listening IP addresses:
  192.168.0.1
  192.168.1.100
  192.168.122.1
Connections:
       host:  192.168.0.1...192.168.0.2  IKEv2
       host:   local:  [C=cz, O=gw1, OU=gw1, CN=gw1] uses public key authentication
       host:    cert:  "C=cz, O=gw1, OU=gw1, CN=gw1"
       host:   remote: [C=cz, O=gw2, OU=gw2, CN=gw2] uses public key authentication
       host:   child:  dynamic === dynamic TUNNEL
Security Associations (1 up, 0 connecting):
       host[1]: ESTABLISHED 82 seconds ago, 192.168.0.1[C=cz, O=gw1, OU=gw1, CN=gw1]...192.168.0.2[C
=cz, O=gw2, OU=gw2, CN=gw2]
       host[1]: IKEv2 SPIs: f80e12a7e08cb059_i* f79498a3eea2da1c_r, public key reauthentication in 4
9 minutes
       host[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
       host{1}:  INSTALLED, TUNNEL, ESP SPIs: cc700665_i c1b8f6c4_o
       host{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 12 minutes
       host{1}:   192.168.0.1/32 === 192.168.0.2/32
```

Obrázek 4.30: Parametry sestaveného spojení host-to-host-cert – ipsec statusall

Výpis podrobností certifikátu certifikační autority příkazem `ipsec listcacerts` je na obrázku 4.31.

```
root@ubuntu:/home/radim# ipsec listcacerts

List of X.509 CA Certificates:

  subject:  "C=cz, O=ca, OU=ca, CN=CA_strongswan"
  issuer:   "C=cz, O=ca, OU=ca, CN=CA_strongswan"
  serial:    01
  validity:  not before Feb 13 13:09:00 2014, ok
             not after  Feb 13 13:09:00 2024, ok
  pubkey:    RSA 2048 bits
  keyid:     f9:ff:d1:bd:56:24:70:02:1f:b1:13:9d:1e:a3:8e:06:18:c2:9d:6a
  subjkey:   d5:0b:4a:66:32:27:7c:62:c3:0f:b5:09:86:34:2a:82:35:ba:5e:1d
  authkey:   d5:0b:4a:66:32:27:7c:62:c3:0f:b5:09:86:34:2a:82:35:ba:5e:1d
```

Obrázek 4.31: Výpis podrobnosti certifikátu certifikační autority CA_strongswan

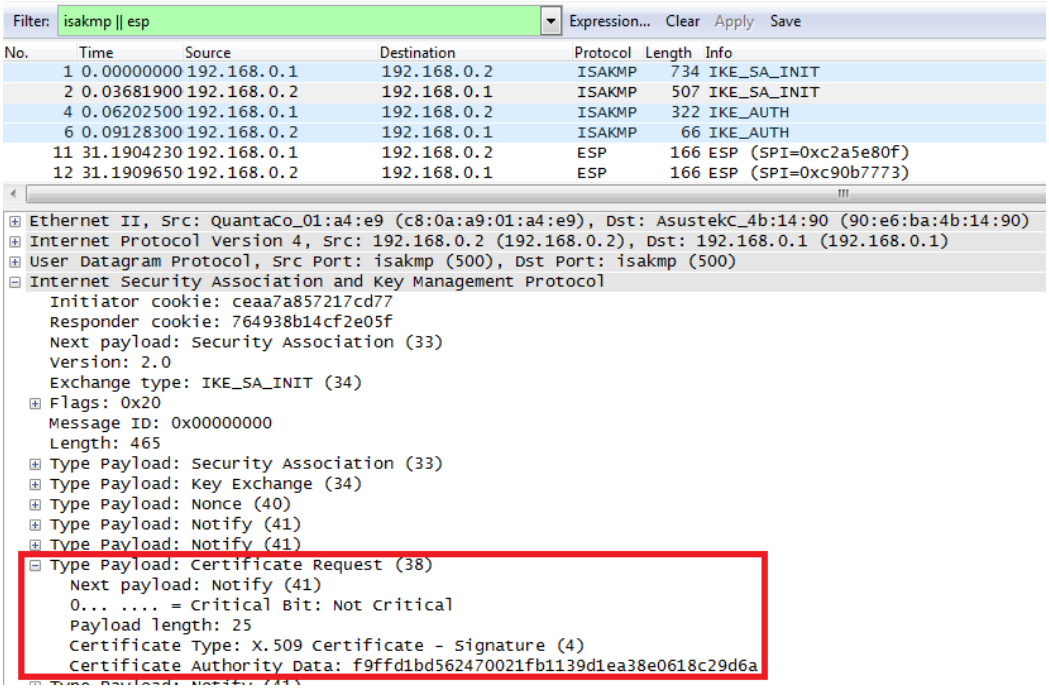
Výpis certifikátů obou hostů příkazem `ipsec listcerts` je znázorněn na obrázku 4.32

```
root@ubuntu:/home/radim# ipsec listcerts

List of X.509 End Entity Certificates:

  subject:  "C=cz, O=gw1, OU=gw1, CN=gw1"
  issuer:   "C=cz, O=ca, OU=ca, CN=CA_strongswan"
  serial:    03
  validity:  not before Feb 13 13:19:00 2014, ok
             not after  Feb 13 13:19:00 2015, ok
  pubkey:    RSA 2048 bits, has private key
  keyid:     dc:a0:d0:09:be:ce:02:2d:df:27:fa:26:d2:79:80:22:ed:ff:16:a3
  subjkey:   60:42:af:8b:38:92:cf:4c:4c:15:2b:3e:25:67:fa:c6:ff:fd:94:d1
  authkey:   d5:0b:4a:66:32:27:7c:62:c3:0f:b5:09:86:34:2a:82:35:ba:5e:1d

  subject:  "C=cz, O=gw2, OU=gw2, CN=gw2"
  issuer:   "C=cz, O=ca, OU=ca, CN=CA_strongswan"
  serial:    04
  validity:  not before Feb 13 13:20:00 2014, ok
             not after  Feb 13 13:20:00 2015, ok
  pubkey:    RSA 2048 bits
  keyid:     2e:19:83:2d:76:61:44:99:9f:e5:b9:cb:86:cf:56:ae:ca:27:ef:cf
  subjkey:   44:10:2d:bd:89:24:19:c0:c5:3b:99:67:54:0d:7a:8d:cb:32:f8:4f
  authkey:   d5:0b:4a:66:32:27:7c:62:c3:0f:b5:09:86:34:2a:82:35:ba:5e:1d
```

Obrázek 4.32: Výpis podrobností certifikát u spojení host-host-cert

Vytvoření provozu je provedeno opět příkazem ping z hosta s IP adresou 192.168.0.1 na IP adresu 192.168.0.2.

Provoz zachycený programem Wireshark (obrázek 4.33). V červeném obdélníku je vyznačena žádost o autentizaci pomocí certifikátů. Je zde uveden typ certifikátu tj. X.509 a data certifikační autority, která certifikát vydala.



Obrázek 4.33: Zachycení provozu programem Wireshark – host-host-cert

4.2.3 Vzdálený přístup s využitím klientského certifikátu pro autentizaci

Tato realizace, která je znázorněna na obrázku 4.34, poskytuje konfiguraci vzdáleného klienta (roadwarrior) pro připojení k bráně GW a přístup do vnitřní sítě `10.1.0.0`. Autentizace je založena na certifikátu brány (`gw.crt`) a certifikátu vzdáleného klienta (`klient.crt`). Oba certifikáty jsou digitálně podepsány dříve vytvořenou certifikační autoritou `CA-strongswan`. Sestavené spojení je otestováno příkazem `ping` vzdáleného klienta, do sítě `10.1.0.0` na klienta Alice (IP adresa 10.1.0.10).

Nastavení síťových rozhraní a pravidel iptables:

Zapnutí směrování, neboli předávání paketů z jednoho síťového rozhraní na druhé:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

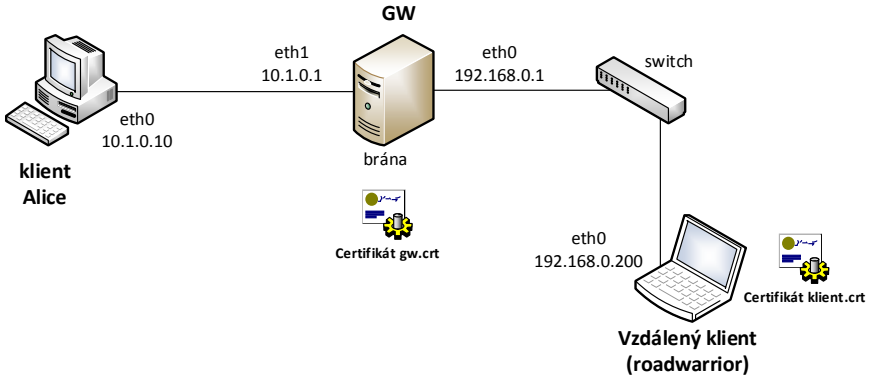
Provedené nastavení zkontrolujeme následujícím příkazem:

```
cat /proc/sys/net/ipv4/ip_forward
```

Pokud je routování zapnuto vypíše 1, pokud je vypnuto vypíše 0.

Nastavení pravidel IPTABLES na bráně GW.

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
```



Obrázek 4.34 Schéma konfigurace vzdáleného klienta s autentizací pomocí certifikátu

GW – konfigurační soubory

 # /etc/ipsec.conf - strongSwan IPsec configuration file

```
conn                                                    %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev2
conn                                                    home
      left=192.168.0.1
      leftcert=gw.crt       #certifikát brány GW
      leftid=gw
      leftsubnet=10.1.0.0/24
      leftfirewall=yes
      right=%any            #jakákoliv vzdálená IP adresa se může
připojit
      auto=add
```

 # /etc/ipsec.secrets - strongSwan IPsec secrets file

```
: RSA gwkey.pem         #soukromý klíč hosta (gwkey)
```

 # /etc/strongswan.conf - strongSwan configuration file

```
charon {
 load = curl test-vectors aes des sha1 sha2 md5 pem pkcs1 pkcs8 gmp
```

```
random nonce x509 revocation hmac xcbc cmac ctr ccm gcm stroke
kernel-netlink socket-default updown  }     #načtení modulů IKE
démona strongSwan
```

Vzdálený klient – konfigurační soubory

 # /etc/ipsec.conf - strongSwan IPsec configuration file

```
conn %default
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev2
conn home
      left=192.168.0.100
      leftcert=klient.crt   #certifikát klienta
      leftid=klient         #"SubjectAltName" certifikátu klient
      leftfirewall=yes
      right=192.168.0.1
      rightid=gw
      rightsubnet=10.1.0.0/24
      auto=add              #načte spojení při startu IKE démona bez
spuštění
```

 # /etc/ipsec.secrets - strongSwan IPsec secrets file

```
: RSA klientKey.pem  #soukromý klíč klienta
```

 # /etc/strongswan.conf - strongSwan configuration file

```
charon {
 load = curl test-vectors aes des sha1 sha2 md5 pem pkcs1 pkcs8 gmp
random nonce x509 revocation hmac xcbc cmac ctr ccm gcm stroke
kernel-netlink socket-default updown #načtené moduly ike démona
}
```

Na bráně GW a Vzdáleném klientovi je spuštěn strongSwan. Sestavení spojení je provedeno na Vzdáleném klientovi příkazem `ipsec up home`. Výpis o sestavení spojení je znázorněn na obrázku 4.35

```
root@ubuntu:/home/radim# ipsec up home
retransmit 4 of request with message ID 0
sending packet: from 192.168.0.100[500] to 192.168.0.1[500] (708 bytes)
received packet: from 192.168.0.1[500] to 192.168.0.100[500] (465 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(MUL
T_AUTH) ]
received cert request for "C=cz, O=ca, OU=ca, CN=CA_strongswan"
sending cert request for "C=cz, O=ca, OU=ca, CN=CA_strongswan"
authentication of 'C=cz, O=vzd??len?? klient, OU=vzd??len?? klient, CN=klient' (
myself) with RSA signature successful
sending end entity cert "C=cz, O=vzd??len?? klient, OU=vzd??len?? klient, CN=kli
ent"
establishing CHILD_SA home
generating IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi
 TSr N(MOBIKE_SUP) N(NO_ADD_ADDR) N(MULT_AUTH) N(EAP_ONLY) ]
sending packet: from 192.168.0.100[4500] to 192.168.0.1[4500] (1804 bytes)
received packet: from 192.168.0.1[4500] to 192.168.0.100[4500] (1500 bytes)
parsed IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP)
N(ADD_4_ADDR) ]
received end entity cert "C=cz, O=gw1, OU=gw1, CN=gw1"
  using certificate "C=cz, O=gw1, OU=gw1, CN=gw1"
  using trusted ca certificate "C=cz, O=ca, OU=ca, CN=CA_strongswan"
checking certificate status of "C=cz, O=gw1, OU=gw1, CN=gw1"
certificate status is not available
  reached self-signed root ca with a path length of 0
authentication of 'C=cz, O=gw1, OU=gw1, CN=gw1' with RSA signature successful
IKE_SA home[1] established between 192.168.0.100[C=cz, O=vzd??len?? klient, OU=v
zd??len?? klient, CN=klient]...192.168.0.1[C=cz, O=gw1, OU=gw1, CN=gw1]
scheduling reauthentication in 3416s
maximum IKE_SA lifetime 3596s
CHILD_SA home{1} established with SPIs c64b141b_i cf5698e9_o and TS 192.168.0.10
0/32 === 10.1.0.0/24
connection 'home' established successfully
```

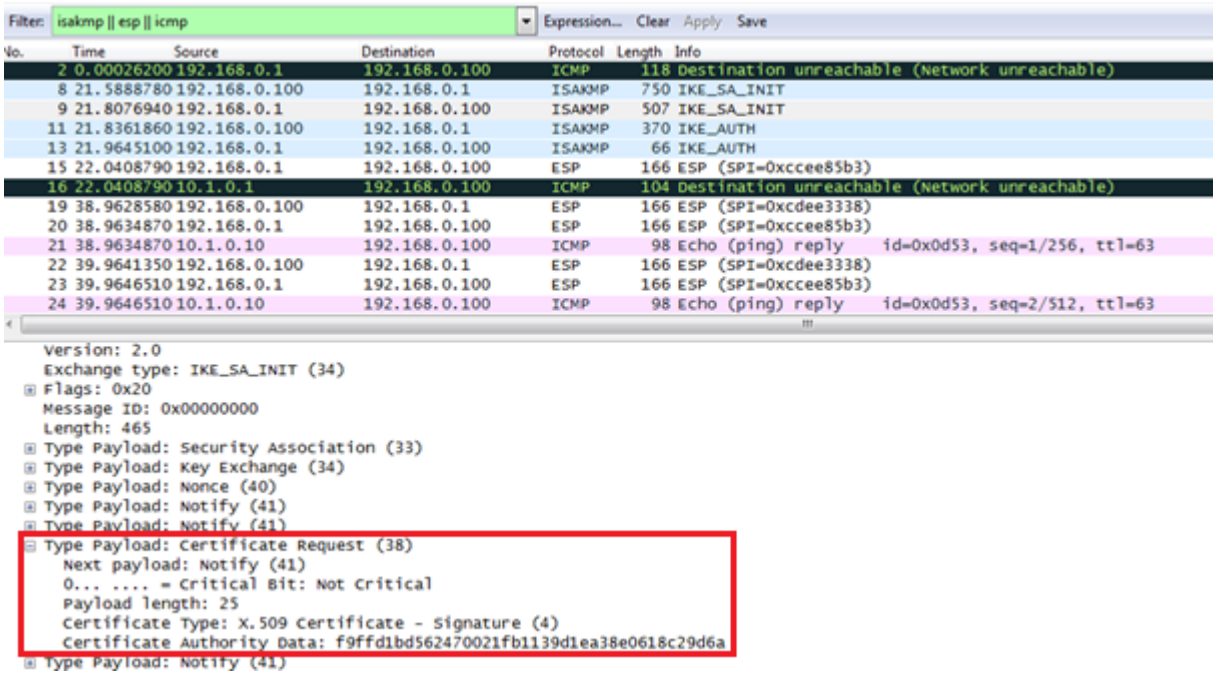
Obrázek 4.35: Sestaveni spojení vzdáleného klienta k bráně GW

Podrobnosti o sestaveném spojení s názvem `home` je uvedeno na obrázku 4.36. Jsou zde např. vidět certifikáty klienta a brány, které jsou zobrazeny ve žlutém rámečku.

```
root@ubuntu:/home/radim# ipsec statusall
Status of IKE charon daemon (strongSwan 5.1.2, Linux 3.11.0-12-generic, x86_64):
  uptime: 3 minutes, since Mar 17 14:40:29 2014
  malloc: sbrk 2297856, mmap 0, used 232784, free 2065072
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
  loaded plugins: charon aes des sha1 sha2 md5 pem pkcs1 pkcs8 gmp random nonce x509 revocation h
mac xcbc cmac stroke kernel-netlink socket-default updown
Listening IP addresses:
  192.168.0.100
Connections:
       home:  192.168.0.100...192.168.0.1  IKEv2
       home:   local:  [C=cz, O=vzd??len?? klient, OU=vzd??len?? klient, CN=klient] uses public
key authentication
       home:    cert:  "C=cz, O=vzd??len?? klient, OU=vzd??len?? klient, CN=klient"
       home:   remote: [C=cz, O=gw1, OU=gw1, CN=gw1] uses public key authentication
       home:   child:  dynamic === 10.1.0.0/24 TUNNEL
Security Associations (1 up, 0 connecting):
       home[1]: ESTABLISHED 2 minutes ago, 192.168.0.100[C=cz, O=vzd??len?? klient, OU=vzd??len?
? klient, CN=klient]...192.168.0.1[C=cz, O=gw1, OU=gw1, CN=gw1]
       home[1]: IKEv2 SPIs: 0805a2b5161ed2d3_i* 5df4096fbc8cea61_r, public key reauthentication
in 49 minutes
       home[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
       home{1}:  INSTALLED, TUNNEL, ESP SPIs: c64b141b_i cf5698e9_o
       home{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 11 minutes
       home{1}:   192.168.0.100/32 === 10.1.0.0/24
```

Obrázek 4.36: Podrobnosti o sestaveném spojení vzdáleného klienta a brány GW

Zachycení provozu na bráně GW programem Wireshark je znázorněno na obrázku 4.37. Ve výpisu jsou zobrazeny pomocí filtru jen protokoly ISAKMP, ESP a ICMP. Uvedený výpis zobrazuje sestavení spojení a zabezpečenou komunikaci protokolem ESP do vnitřní sítě na IP adresu `10.1.0.10`. V červeném rámečku je poté uvedena autentizace pomocí certifikátů a data certifikační autority.



Obrázek 4.37: Zachycení zabezpečeného provozu mezi klientem a branou GW

4.3 IPsec a NAT

NAT je zkratka pro Network Address Translation, tedy překlad IP adres. Upravuje síťový provoz přes směrovač přepisem výchozí nebo cílové IP adresy. Používá se pro přístup více počítačů z lokální sítě na Internet, pod jedinou veřejnou adresou.

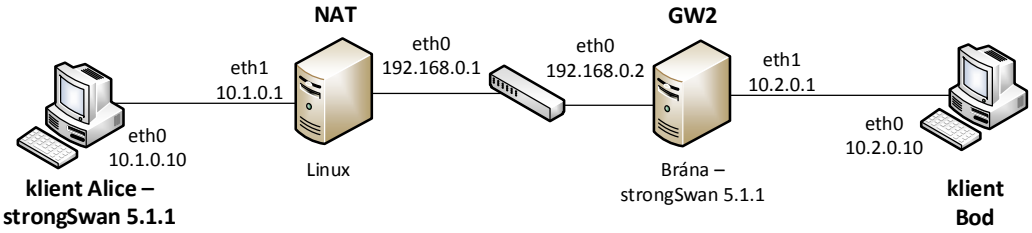
Princip NAT:

1. Klient se připojí na bránu vnitřní sítě
2. Směrovač pakety zachytí, změní jejich IP adresu na adresu z rozsahu své vnitřní sítě
3. Směrovač pakety označí tak, že je odešle z náhodného TCP portu, resp. UDP portu
4. Směrovač si do tabulky zapíše, který port zvolil a který klient k němu patří
5. Při přijetí odpovědi provede směrovač reverzní akci a pakety vrátí klientovi

V tomto scénáři se klient Alice nachází za NAT směrovačem, který v našem případě simuluje PC s linuxem. Na klientu Alice a bráně GW2 máme nainstalován strongSwan ve verzi 5.1.1. Úkolem je vytvořit tunelové spojení na bránu GW2, které prochází přes NAT směrovač. Pro přechod přes tento směrovač je použito UDP zapouzdření paketů. Klient Alice a brána GW2 se navzájem autentizují

pomocí předsdílených klíčů. Vytvořené spojení je otestováno příkazem `ping` z klienta Alice na klienta Boba, který leží za bránou GW2 v podsíti `10.2.0.0`.

Pozn.: V IKEv2 režimu, je u strongSwan defaultně nastavena podpora NAT. Komunikace automaticky probíhá na portu UDP 4500.



Obrázek 4.38:Schéma konfigurace IPsec a NAT

Alice - konfigurační soubory

 # /etc/ipsec.conf - strongSwan IPsec configuration file

```
config setup

conn %default
      keyexchange=ikev2
      authby=secret
      ike=3des-md5-modp1024!      #šifra-has. algoritmus-DH-algoritmus
      esp=3des-md5-modp1024!      #šifra-has. algoritmus-DH-algoritmus

conn nat-t
      left=%defaultroute          #VPN dostupná přes všechna síťová
rozhraní
      leftfirewall=yes
      right=192.168.0.2
      rightsubnet=10.2.0.0/24
      auto=add                    #načte spojení při startu IKE démona
bez spuštění
```

 # /etc/ipsec.secrets - strongSwan IPsec secrets file

```
10.1.0.10 : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL     #soukromý klíč
```

 # /etc/strongswan.conf - strongSwan configuration file

```
charon {
load = aes des sha1 sha2 md5 pem pkcs1 gmp random nonce hmac xcbc
stroke kernel-netlink socket-default updown
}
```

GW2 - konfigurační soubory

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
config setup

conn %default
      keyexchange=ikev2
      authby=secret
      ike=3des-md5-modp1024!      #šifra-has. algoritmus-DH-algoritmus
      esp=3des-md5-modp1024!      #šifra-has. algoritmus-DH-algoritmus

conn nat-t
      left=192.168.0.2
      leftsubnet=10.2.0.0/24      #místní podsíť
      leftfirewall=yes
      right=%any                  #kdokoliv se může připojit
      rightsubnet=10.1.0.0/24     #vzdálená podsíť
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
10.1.0.10 : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL
```

 # /etc/strongswan.conf - strongSwan configuration file

```
charon {
 load = aes des sha1 sha2 md5 pem pkcs1 gmp random nonce hmac xcbc
stroke kernel-netlink socket-default updown
}
```

NAT – Linux - konfigurace

Zde je uvedena konfigurace PC, na kterém je simulováno připojení přes NAT.

Zapnutí přeposílání paketu z jednoho síťového rozhraní na druhé:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nastavení pravidel iptables pro povolení NAT na rozhraní eth0:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Povolení pravidla pro přeposílání paketů z jednoho rozhraní na druhé.

```
iptables -A FORWARD -i eth2 -j ACCEPT
```

Na obrázku 4.39 je uvedeno sestavení spojení s názvem `nat-t`. Ve žlutém rámečku je označena detekce NAT.

```
initiating IKE_SA nat-t[1] to 192.168.0.2
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
sending packet: from 10.1.0.10[500] to 192.168.0.2[500] (692 bytes)
received packet: from 192.168.0.2[500] to 10.1.0.10[500] (440 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(MULT_AUTH)
]
local host is behind NAT, sending keep alives
no IDi configured, fall back on IP address
authentication of '10.1.0.10' (myself) with pre-shared key
establishing CHILD_SA nat-t
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE
_SUP) N(ADD_4_ADDR) N(MULT_AUTH) N(EAP_ONLY) ]
sending packet: from 10.1.0.10[4500] to 192.168.0.2[4500] (396 bytes)
received packet: from 192.168.0.2[4500] to 10.1.0.10[4500] (252 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(ADD
_4_ADDR) N(ADD_4_ADDR) ]
authentication of '192.168.0.2' with pre-shared key successful
IKE_SA nat-t[1] established between 10.1.0.10[10.1.0.10]...192.168.0.2[192.168.0
.2]
scheduling reauthentication in 3419s
maximum IKE_SA lifetime 3599s
connection 'nat-t' established successfully
```

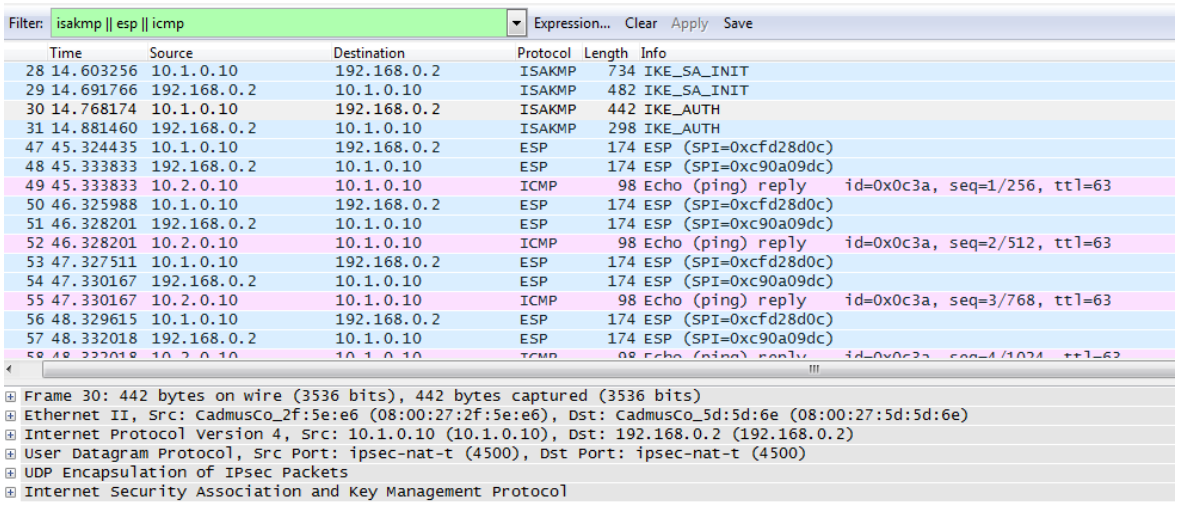
Obrázek 4.39: Výpis sestavení spojení nat-t

Obrázek 4.40 znázorňuje podrobný výpis příkazem `ipsec statusall` právě sestaveného spojení `nat-t`. Ve žlutém rámečku jsou označeny nastavené šifry, haše a DH-algoritmus. Je zde patrný rozdíl oproti předchozím příkladům, kde je využíváno defaultní nastavení (AES128/SHA1/MODP2048).

```
Status of IKE charon daemon (strongSwan 5.1.1, Linux 3.2.0-29-generic, i686):
  uptime: 7 minutes, since Apr 01 02:33:36 2014
  malloc: sbrk 135168, mmap 0, used 82816, free 52352
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 4
  loaded plugins: charon aes des sha1 sha2 md5 pem pkcs1 gmp random nonce hmac xcbc stroke
kernel-netlink socket-default updown
Listening IP addresses:
  10.0.2.15
  10.1.0.10
Connections:
       nat-t:  %any...192.168.0.2  IKEv2
       nat-t:   local:  uses pre-shared key authentication
       nat-t:   remote: [192.168.0.2] uses pre-shared key authentication
       nat-t:   child:  dynamic === 10.2.0.0/24 TUNNEL
Security Associations (1 up, 0 connecting):
       nat-t[1]: ESTABLISHED 7 minutes ago, 10.1.0.10[10.1.0.10]...192.168.0.2[192.168.0.2]
       nat-t[1]: IKEv2 SPIs: a4dbd3fba08187fa_i* eb71934b4b76ceb0_r, pre-shared key reauthe
ntication in 45 minutes
       nat-t[1]: IKE proposal: 3DES_CBC/HMAC_MD5_96/PRF_HMAC_MD5/MODP_1024
       nat-t{1}:  INSTALLED, TUNNEL, ESP in UDP SPIs: c769089b_i c1aae38f_o
       nat-t{1}:  3DES_CBC/HMAC_MD5_96, 0 bytes_i, 0 bytes_o, rekeying in 8 minutes
       nat-t{1}:   10.1.0.10/32 === 10.2.0.0/24
```

Obrázek 4.40: Podrobný výpis spojení nat-t

Obrázek 4.41 zachycuje provoz dle uvedeného schématu s NAT.



Obrázek 4.41: Zachycení provozu u spojení nat-t programem wireshark

5 Použití certifikátů v USB tokenech

Tato část práce bude zaměřena na popis USB Tokenů za použití softwaru strongSwan. USB Token je hardwarové zařízení, které slouží k uložení důvěrných dat, jako jsou digitální certifikáty, veřejné a privátní klíče. Takto bezpečně uložena data nelze neautorizovaně využívat. V praxi se poté token využívá jako elektronický klíč, který se pomocí USB portu připojí k počítači. Informace uložené na tokenu slouží k ověření daného uživatele. Většinou je nutno po připojení zadat ještě PIN, který data ochrání při ztrátě nebo odcizení zařízení.

Použitý token pro realizaci (obrázek 5.1): Aladdin USB eToken Pro 32k (4.2 B)

Technické specifikace použité tokenu viz [14].

Obrázek 5.1: USB Token Aladdin Pro 32k

5.1 Instalace pod operačním systémem Linux

Pod novějšími operačními systémy Linux není tento USB token podporován. Starší operační systémy nemají funkční balíčkovací systém (výrobce ukončil podporu pro tyto systémy), tudíž není možné doinstalovat jakékoliv balíčky a knihovny pro přístup na USB Token Aladdin Pro 32k. Aplikace etoken PKI client pro Ubuntu ve verzi 5.00, je poslední vydanou a je z roku 2009. Při instalaci jí instalační program systému Ubuntu označí jako nedůvěryhodnou a instalace neproběhne v pořádku. Z toho vyplývá, že není možné potřebné knihovny pro tento token nainstalovat. Uvedené balíčky jsou pro instalaci nezbytné.

Potřebné balíčky k instalaci:

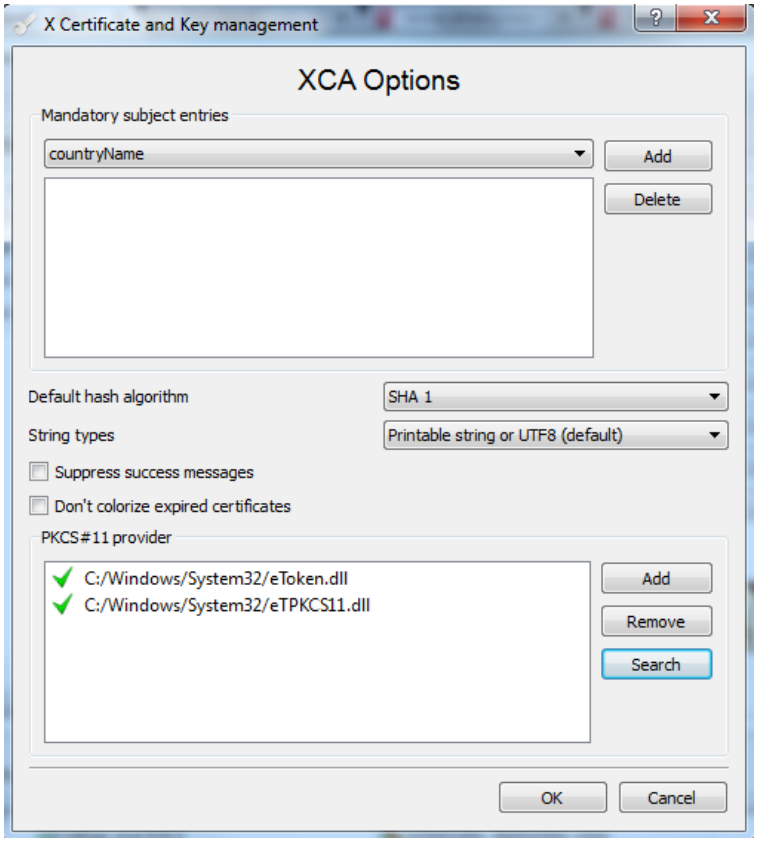
```
apt-get install libqt4-core libqt4-gui pcscd opensc
libqt4-core a libqt4-gui - balíčky GUI runtime knihovny
pcscd – smartcard démon
opensc – software pro čipové karty
```

5.2 Instalace pod operačním systémem Windows

Provede se instalace aplikace eToken PKI Client (v tomto případě eToken PKI Client 5.1 SP1), která slouží pro instalaci ovladačů tokenu. Token je pod systémem Windows funkční a lze na něj importovat certifikáty a klíče. Aplikace je k dispozici na adrese viz [16].

5.3 Inicializace tokenu v programu XCA

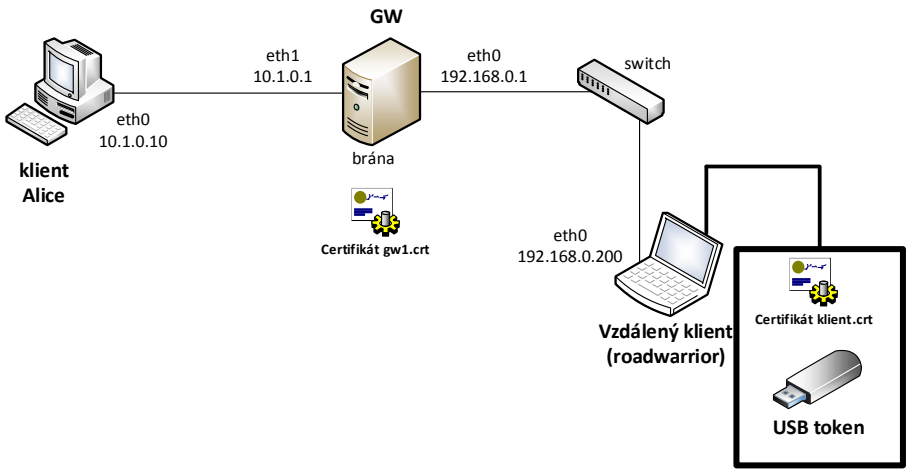
Po instalaci PKI klienta je v programu XCA na záložce „File“, možnost „Options“ kliknuto na tlačítko Start a začne vyhledávání potřebných knihoven .dll (jsou nainstalovány s programem eToken PKI Client a umístěny ve složce C:/Windows/System32) pro inicializaci tokenu v programu, které nám ukazuje obrázek 5.2. Vyhledané soubory vybereme a potvrdíme stiskem „Ok“. Tímto byla provedena inicializace tokenu v programu XCA a nyní již můžeme nastavit heslo, které nám slouží pro ochranu a import klíčů a certifikátů na token. Všechny tyto operace jsou již nyní dostupné v záložce „Token“ v hlavním okně programu. Pro import certifikátu nebo soukromého klíče, stačí na něj kliknout pravým tlačítkem a zvolit možnost „Store on Security token“. Po vyplnění nastaveného hesla je klíč nebo certifikát uložen na token.



Obrázek 5.2: Inicializace potřebných .dll souborů tokenu

5.4 Konfigurace s využitím USB tokenu

V této kapitole bude teoreticky uvedeno, jak se konfiguruje strongSwan pro práci s USB tokeny. Je zde uvedena vzorová konfigurace, která ovšem nebyla prakticky otestována s důvodu nekompatibility použitého tokenu s novějšími operačními systémy Linux, jak bylo uvedeno výše. Obrázek 5.3 znázorňuje schéma vzdáleného přístupu za použití USB tokenu. Uvedená konfigurace je pro strongSwan ve verzi 4.3.2.



Obrázek 5.3: Schéma pro využití USB tokenu a jeho certifikátu

Konfigurační soubory brány GW

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
config setup
      charonstart=no #vypnuti IKE démona charon
      plutostart=yes #start IKE démona pluto

conn %default          #parametry spojení
      ikelifetime=60m
      keylife=20m
      rekeymargin=3m
      keyingtries=1
      keyexchange=ikev1      #verze ike protokolu
conn home
      left=%defaultroute     #automatické nastavení adresy
      leftcert=gw.crt        #certifikát brány
      leftsubnet=10.1.0.0/24      #podsíť za branou GW
      leftfirewall=yes
      right=%any
      auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
: RSA moonKey.pem      #soukromý klíč brány
```

#/etc/strongswan.conf - strongSwan configuration file

```
pluto {
```

```
    load = curl test-vectors aes des sha1 sha2 md5 pem pkcs1 gmp
random x509 hmac xcbc stroke kernel-netlink
    updown
    }     #moduly ike démona pluto
```

Konfigurace klienta

#/etc/ipsec.conf - strongSwan IPsec configuration file

```
config setup
    plutostart=yes
    pkcs11module = /usr/lib/opensc-pkcs11.so #načtení modulů pro
token

conn %default   #defaultní nastavení není povinné
    ikelifetime=60m
    keylife=20m
    rekeymargin=3m
    keyingtries=1
    keyexchange=ikev1

conn home       #název spojení
    left=%defaultroute
    leftcert=%smartcard   #certifikát klienta načítán z tokenu
    leftfirewall=yes
    right=192.168.0.1     #IP adresa rozhraní brány
    rightid=gw            #identifikátor certifikátu brány
(SubjectAltName)
    rightsubnet=10.1.0.0/24     #podsíť do které se připojuje
    auto=add
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
    : PIN %smartcard %prompt #zadání PIN kódu pro přístup na USB
token
```

```
#/etc/strongswan.conf - strongSwan configuration file
pluto {
      load = curl test-vectors aes des sha1 sha2 md5 pem pkcs1 gmp
random x509 hmac xcbc stroke kernel-netlink
      updown
      }
```

Tato kapitola alespoň teoreticky nastínila jak strongSwan pracuje s USB tokeny.

6 StrongSwan v mobilních telefonech

Mobilních operačních systémů již dnes existuje několik a s tím i vysoký počet uživatelů, kteří stále více mobilní zařízení využívají. StrongSwan neopomíjí tento široký okruh uživatelů a zaměřuje se na využití svého VPN klienta v těchto zařízeních.

6.1 VPN klient pro Android 4.x

Klient strongSwan VPN pro Android 4.x lze stáhnout přímo z Google Play [15]. Jeho poslední verze je 1.3.3, tento klient je pouze pro operační systém Android, pro jiné mobilní operační systémy k dispozici není.

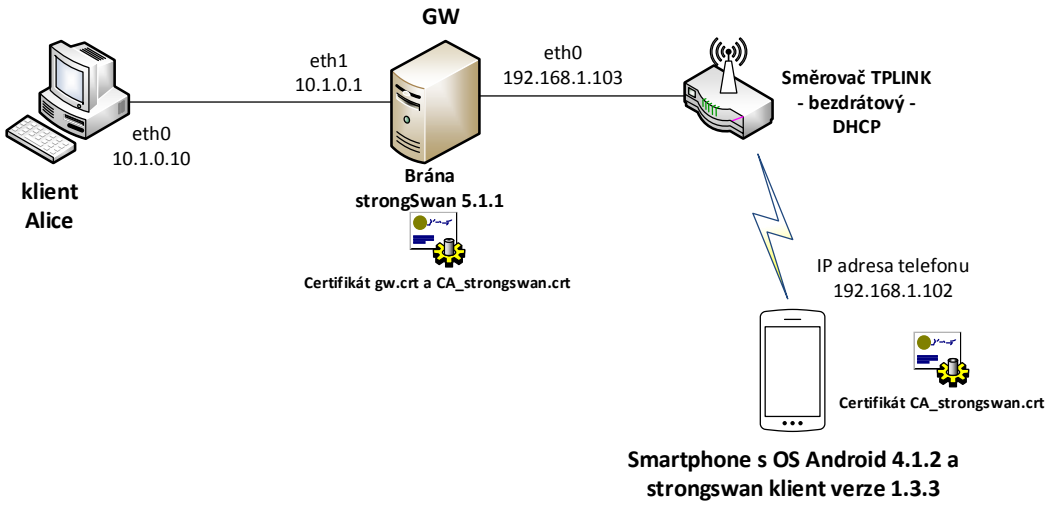
Vlastnosti strongSwan VPN klienta:

- Použití na operačních systémech Android 4 a novějších
- Využívá IKEv2 protokol (IKEv1 není podporován)
- Autentizace uživatelů je ověřována buď pomocí EAP na základě uživatelského jména nebo hesla (EAP-MSCHAPv2, EAP-MD5, EAP-GTC), nebo RSA autentizací pomocí soukromého klíče nebo certifikátu
- Podpora šifer AES-CBC, AES-GCM a hašovacích algoritmů SGA1/SHA2

6.2 Konfigurace s využitím klienta strongSwan v mobilním telefonu

V této konfiguraci je použit protokol EAP-MSCHAPv2, patřící pod třídu protokolů PEAPv0. Široce se využívá i u Windows 7 VPN klienta. Autentizace klienta probíhá nejprve na základě certifikátu certifikační autority a certifikátu brány GW, po úspěšném ověření certifikátů klient posílá své uživatelské jméno a heslo pro ověření.

Na obrázku 6.1 je uvedeno celé schéma testované konfigurace. Bezdrátový směrovač TPLINK přidělí bráně GW a smartphonu IP adresy (DHCP). Na bráně GW je spuštěn strongSwan 5.1.1, jeho konfigurační soubory jsou uvedeny níže. Dále jsou zde uloženy certifikáty `CA_strongswan.crt` a `gw.crt`. Certifikát certifikační autority je uložen i na smartphonu s operačním systémem Android. Tento operační systém je ve verzi 4.1.2.



Obrázek 6.1: Schéma pro připojení smartphonu s klientem strongSwan

Pro použití protokolu MSCHAPv2 musíme provést kompilaci programu strongSwan s následujícím nastavením:

```
./configure --prefix=/usr --sysconfdir=/etc --enable-md4 --enable-eap-mschapv2 --enable-openssl --enable-eap-identity
```

Tím aktivujeme dané moduly pro otestování navržené konfigurace.

Následuje instalace příkazy:

```
make
make install
```

Umístění a instalace certifikátů:

- **Na bráně GW**

Certifikát `CA_strongswan.crt` do složky cacerts, certifikát `gw.crt` do složky certs. Soukromý klíč `gwKey.pem` certifikátu `gw.crt` do složky private. U certifikátu brány `gw.crt` je důležité, aby alternativní název certifikátu (SubjectAltName) byl stejný, jako brána ke které se připojuje (v tomto případě 192.168.1.103).

- **Ve smartphonu**

Operační systém Android podporuje certifikáty X.509 s kódováním DER uložené jako soubory s příponou CRT nebo CER.

Certifikát certifikační autority `CA_strongswan.crt` je nakopírován do kořenového adresáře interního uložiště telefonu a jeho instalace je následující:

1. V Nastavení > Osobní > Zabezpečení > Úložiště pověření > Instalovat z úložiště - je vybrán.
2. Poté je nainstalován.

Konfigurační soubory brány GW:

```
#/etc/ipsec.conf - strongSwan IPsec configuration file
conn %default                  #defaultní nastavení
      keyexchange=ikev2        #protokol IKEv2
      ike=aes256-sha1-modp1024! #šifry
      esp=aes256-sha1!         #šifry a haše esp protokolu
      dpdaction=clear          #vymazání neaktivního spojení
      dpddelay=300s           #po 300 sekundách nečinnosti detekuje aktivitu spojení

conn android      #název spojení
      left=%any  #připojení k bráně VPN přes jakékoliv síťové rozhraní
      leftsubnet=10.1.0.0/24      #podsíť za bránou GW
      leftid=192.168.1.103        #identifikátor brány (SubjectAltName certifikátu)
      leftcert=gw.crt             #certifikát brány
      leftauth=pubkey             #autentizace veřejným klíčem certifikátu
      right=%any                  #jakákoliv vzdálená IP adresa se může připojit
```

```
      rightauth=eap-mschapv2      #uživatele jsou ověřování pomocí
protokolu EAP-MSCHAPv2
      rightsendcert=never   #brána nebude vyžadovat certifikát,
klienti se ověřují pomocí MSCHAPv2
      eap_identity=%any     #ověření eap identity
      auto=add              #načtení démona charon a čekání na
klienty
```

#/etc/ipsec.secrets - strongSwan IPsec secrets file

```
: RSA gwKey.pem             #soukromý klíč brány
android : EAP "klient"      #uživatelské jméno : EAP „heslo"
```

 # /etc/strongswan.conf - strongSwan configuration file

```
charon {
      load = curl aes des sha1 sha2 md4 md5 pem pkcs1 gmp random
nonce x509 revocation hmac xcbc stroke kernel-netlink socket-default
fips-prf eap-mschapv2 eap-identity updown
      #načtení potřebných modulů IKE démona charon
```

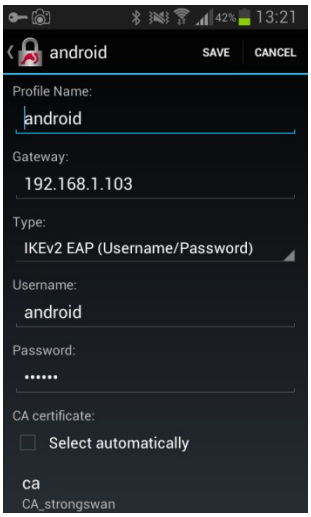
Připojení VPN klienta strongSwan na bránu GW:

Po spuštění aplikace ve smartphonu je zobrazen konfigurační formulář pro vytvoření nového připojení (obrázek 6.2).

Jednotlivé položky konfigurace:

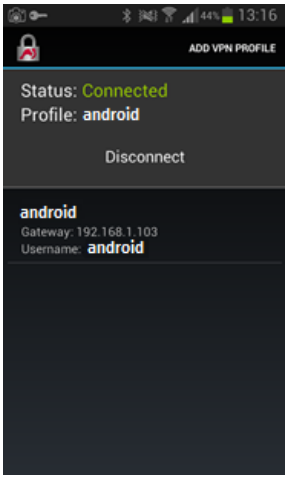
- Název profilu (Profile Name): **android**
- Název brány, nebo IP adresa (Gateway): **192.168.1.103**
- Typ připojení (Type): **IKEv2 EAP**
- Uživatelské jméno (Username): **android**
- Heslo (Password): **klient**
- Certifikát certifikační autority (CA certificate): **certifikát ca_strongswan, který byl instalován**

Nastavená konfigurace je uložena tlačítkem SAVE vpravo nahoře.



Obrázek 6.2: Nastavení parametrů spojení s názvem android

Aktivní a úspěšně připojení znázorňuje obrázek 6.3.

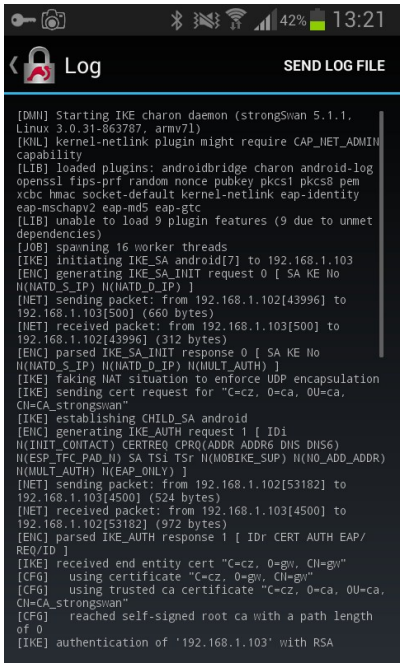


Obrázek 6.3: Úspěšně vytvořené spojení klienta –Status: Connected

V notifikační liště OS Android je zobrazeno v červeném rámečku logo klíče (obrázek 6.4), které znázorňuje úspěšně připojení. Po otevření je vidět právě vytvořená relace android s dobou trvání, odeslanými a přijatými pakety. V této notifikaci je možno spojení ukončit, nebo provést jeho konfiguraci.

Obrázek 6.4: Běžící spojení VPN klienta strongSwan

U klienta je možné zobrazit kompletní log soubor vytvoření spojení a ten poslat (SEND LOG FILE – vpravo nahoře), např. pomocí Bluetooth nebo emailu. Log soubor je uveden na obrázku 6.5.



Obrázek 6.5: Log soubor vytvořený na klientovi

Kompletní log soubor VPN klienta o průběhu sestavení spojení, exportovaný do textového souboru:

```
00[DMN] Starting IKE charon daemon (strongSwan 5.1.1, Linux 3.0.31-863787, armv7l)
00[KNL] kernel-netlink plugin might require CAP_NET_ADMIN capability
00[LIB] loaded plugins: androidbridge charon android-log openssl fips-prf random nonce pubkey
pkcs1 pkcs8 pem xcbc hmac socket-default kernel-netlink eap-identity eap-mschapv2 eap-md5
00[LIB] unable to load 9 plugin features (9 due to unmet dependencies)
00[JOB] spawning 16 worker threads
16[IKE] initiating IKE_SA android[4] to 192.168.1.103
```

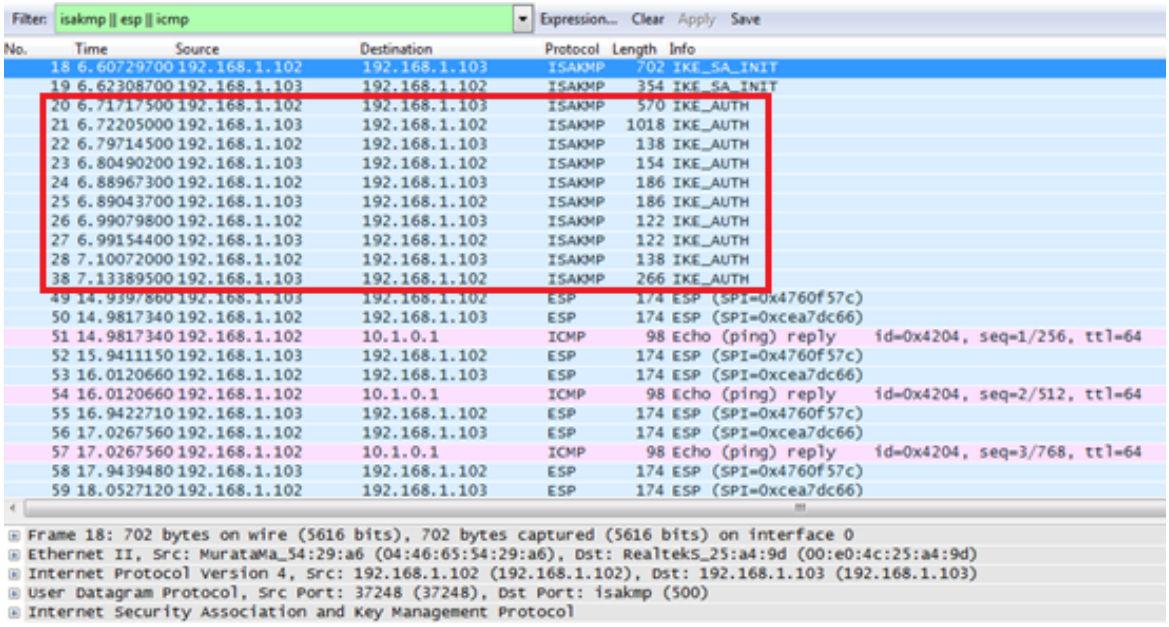
```
16[ENC] generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) ]
16[NET] sending packet: from 192.168.1.102[47023] to 192.168.1.103[500] (660 bytes)
13[NET] received packet: from 192.168.1.103[500] to 192.168.1.102[47023] (312 bytes)
13[ENC] parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(MULT_AUTH) ]
13[IKE] faking NAT situation to enforce UDP encapsulation
13[IKE] sending cert request for "C=cz, O=ca, OU=ca, CN=CA_strongswan"
13[IKE] establishing CHILD_SA android 13[ENC] generating IKE_AUTH request 1 [ IDi
N(INIT_CONTACT) CERTREQ CPRQ(ADDR ADDR6 DNS DNS6) N(ESP_TFC_PAD_N) SA TSi TSr N(MOBIKE_SUP)
N(NO_ADD_ADDR) N(MULT_AUTH) N(EAP_ONLY) ]
13[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (524 bytes)
10[IKE] retransmit 1 of request with message ID 1
10[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (524 bytes)
11[NET] received packet: from 192.168.1.103[4500] to 192.168.1.102[50239] (972 bytes)
11[ENC] parsed IKE_AUTH response 1 [ IDr CERT AUTH EAP/REQ/ID ]
11[IKE] received end entity cert "C=cz, O=gw, CN=gw"
11[CFG] using certificate "C=cz, O=gw, CN=gw"
11[CFG] using trusted ca certificate "C=cz, O=ca, OU=ca, CN=CA_strongswan"
11[CFG] reached self-signed root ca with a path length of 0
11[IKE] authentication of '192.168.1.103' with RSA signature successful
11[IKE] server requested EAP_IDENTITY (id 0x00), sending 'android'
11[ENC] generating IKE_AUTH request 2 [ EAP/RES/ID ]
11[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (92 bytes)
06[NET] received packet: from 192.168.1.103[4500] to 192.168.1.102[50239] (108 bytes)
06[ENC] parsed IKE_AUTH response 2 [ EAP/REQ/MSCHAPV2 ]
06[IKE] server requested EAP_MSCHAPV2 authentication (id 0x30)
06[ENC] generating IKE_AUTH request 3 [ EAP/RES/MSCHAPV2 ]
06[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (140 bytes)
07[NET] received packet: from 192.168.1.103[4500] to 192.168.1.102[50239] (140 bytes)
07[ENC] parsed IKE_AUTH response 3 [ EAP/REQ/MSCHAPV2 ]
07[IKE] EAP-MS-CHAPv2 succeeded: 'Welcome2strongSwan'
07[ENC] generating IKE_AUTH request 4 [ EAP/RES/MSCHAPV2 ]
07[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (76 bytes)
16[NET] received packet: from 192.168.1.103[4500] to 192.168.1.102[50239] (76 bytes)
16[ENC] parsed IKE_AUTH response 4 [ EAP/SUCC ]
16[IKE] EAP method EAP_MSCHAPV2 succeeded, MSK established
16[IKE] authentication of 'android' (myself) with EAP
16[ENC] generating IKE_AUTH request 5 [ AUTH ]
16[NET] sending packet: from 192.168.1.102[50239] to 192.168.1.103[4500] (92 bytes)
09[NET] received packet: from 192.168.1.103[4500] to 192.168.1.102[50239] (220 bytes)
09[ENC] parsed IKE_AUTH response 5 [ AUTH CPRP(ADDR) SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) ]
09[IKE] authentication of '192.168.1.103' with EAP successful
09[IKE] IKE_SA android[4] established between
192.168.1.102[android]...192.168.1.103[192.168.1.103]
09[IKE] scheduling rekeying in 35653s
09[IKE] maximum IKE_SA lifetime 36253s
09[IKE] installing new virtual IP 192.168.1.102
09[IKE] CHILD_SA android{3} established with SPIs c03f73dc_i c647a423_o and TS
192.168.1.102/32 === 10.1.0.0/24
09[DMN] setting up TUN device for CHILD_SA android{3}
09[DMN] successfully created TUN device
09[IKE] peer supports MOBIKE
```

Výpis příkazem `ipsec statusall` na bráně GW ukazuje obrázek 6.6. Žlutým rámečkem jsou zvýrazněny důležité informace o vytvořené IPsec SA.

```
root@gw-KT600-8237:/home/gw/Stažené/strongswan-5.1.2# ipsec statusall
Status of IKE charon daemon (strongSwan 5.1.2, Linux 3.11.0-12-generic, i686):
  uptime: 2 minutes, since Apr 16 13:53:36 2014
  malloc: sbrk 405504, mmap 0, used 113632, free 291872
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
  loaded plugins: charon aes des sha1 sha2 md4 md5 pem pkcs1 gmp random nonce x509
evocation hmac xcbc stroke kernel-netlink socket-default fips-prf eap-mschapv2 eap-
dentity updown
Virtual IP pools (size/online/offline):
  192.168.1.102: 1/1/0
Listening IP addresses:
  192.168.1.103
  10.1.0.1
Connections:
     android:  %any...%any  IKEv2, dpddelay=300s
     android:   local:  [192.168.1.103] uses public key authentication
     android:    cert:  "C=cz, O=gw, CN=gw"
     android:   remote: uses EAP_MSCHAPV2 authentication with EAP identity '%any'
     android:   child:  10.1.0.0/24 === dynamic TUNNEL, dpdaction=clear
Security Associations (1 up, 0 connecting):
     android[2]: ESTABLISHED 98 seconds ago, 192.168.1.103[192.168.1.103]...192.168
1.102[android]
     android[2]: IKEv2 SPIs: 666b4544b3cda759_i 493cc18649801974_r*, rekeying disab
ed
     android[2]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
     android{2}:  INSTALLED, TUNNEL, ESP in UDP SPIs: cea7dc66_i 4760f57c_o
     android{2}:  AES_CBC_256/HMAC_SHA1_96, 336 bytes_i (4 pkts, 87s ago), 336 byte
_o (4 pkts, 87s ago), rekeying disabled
     android{2}:  10.1.0.0/24 === 192.168.1.102/32
```

Obrázek 6.6: Výpis parametrů vytvořené IPsec SA na bráně GW

Zachycení vytvořeného spojení a provozu programem Wireshark, znázorňuje obrázek 6.7 Autentizace prostřednictvím protokolu ISAKMP probíhá v 5-ti fázích. Poté již probíhá mezi klientem a bránou komunikace, která je zabezpečená protokolem ESP.



Obrázek 6.7: Zachycení vytváření spojení a provozu na bráně GW programem Wireshark

Závěr

Cílem této práce bylo poskytnout přehled o IPsec VPN sítích, otestovat a prakticky realizovat různá řešení IPsec VPN sítí. K tomuto účelu jsem použil program strongSwan, který má velmi velké využití a jeho možnosti, které se týkají zabezpečení síťového provozu na síťové vrstvě jsou obrovské. Proto mohu tento software doporučit jak pro malé nasazení v menších sítích, tak i pro komplexní řešení rozsáhlých sítí, které vyžaduje vysoký stupeň zabezpečení.

V mé práci jsem otestoval a zdokumentoval všechna zabezpečení sítě, která strongSwan nabízí. U každého příkladu jsem vytvořil schéma s popisem a adresováním jednotlivých prvků sítě. Každý konfigurační soubor je doplněn o komentáře, které objasňují daný příkaz v konfiguraci. Dále jsem uvedl podrobné výpisy sestaveného spojení, kde jsou uvedeny všechny parametry a vlastnosti spojení. Každá konfigurace je vždy na závěr doplněna o výpis z programu Wireshark, kterým jsem zachytával generovaný provoz v síti.

Co se týká jednotlivých konfigurací, největší problém nastal u použití USB tokenů. Kdy token, který jsem měl k dispozici nepodporují novější operační systémy Linux a tudíž jsem nemohl prakticky ověřit jejich využití. Nicméně jsem tuto problematiku teoreticky zpracoval a uvedl příklad konfigurace, kde a jak můžeme takový token použít. Ostatní konfigurace probíhali více méně bez problémů. Pro vytváření a práci s certifikáty, dobře posloužil program XCA. U vytváření certifikátů je lepší definovat jeho „SubjectAltName“, kdy tento krok usnadnil poté vzájemnou autentizaci a přehlednost v konfiguračních souborech. StrongSwan neopomíjí i své využití v mobilních telefonech pro operační systém Android. Pro tuto platformu existuje VPN klient, kterého jsem otestoval pomocí konfigurace uvedené v kapitole 6. Ověření uživatele proběhlo pomocí protokolu MSCHAPv2 na základě certifikátu certifikační autority a uživatelského jména a hesla.

Přínos, který má pro mě tato práce je velký. Obohatila mě o vědomosti, které se týkají zabezpečení dat při přenosu v síti a síťování v systému Linux. Dalším přínosem jsou přehledně zpracovány jednotlivé konfigurace, které můžou sloužit jako podklady pro implementování daných sítových řešení. Tato práce tak poskytuje ucelený přehled, který pomůže k lepšímu a rychlejšímu nastudování dané problematiky a práce s tímto programem pro zabezpečení přenosu na síťové vrstvě. Tento program je neustále vyvíjen a s novými verzemi se zdokonalují konfigurace a možnosti využití.

Použitá literatura

[1] *Bezpečnost sítí: velká kniha*. Vyd. 1. Brno: CP Books, 2005, 589 s. ISBN 80-251-0697-7

[2] *Zabezpečení multimediálního přenosu dat v reálném čase.* [online]. [cit. 2014-01-12]. Dostupné z: http://vosec.wz.cz/index.php?stav=ipsec

[3] *Úvod do strongSwan.* [online]. [cit. 2014-01-23]. Dostupné z: http://wiki.strongswan.org/projects/strongswan/wiki/IntroductionTostrongSwan

[4] *Instalace* strongSwan [online]. [cit. 2014-01-23]. Dostupné z: http://wiki.strongswan.org/projects/strongswan/wiki/InstallationDocumentation

[5] *Digitální certifikát.* [online]. [cit. 2014-01-26]. Dostupné z: http://interval.cz/clanky/co-to-je-digitalni-certifikat/

[6] *The strongSwan Open Source VPN Project.* [online]. [cit. 2014-01-30]. Dostupné z: http://www.strongswan.org/docs/dfn_berlin_2011.pdf

[7] MACHNÍK, Petr. Širokopásmové sítě: přednášky. Ostrava, 2013.

[8] THOMAS, Thomas M. *Zabezpečení počítačových sítí bez předchozích znalostí*. Brno: CP Books, 2005, 338 s. ISBN 80-251-0417-6.

[9] *StrongSwan VPN klien.t* [online]. [cit. 2014-02-18]. Dostupné z: https://play.google.com/store/apps/details?id=org.strongswan.android&hl=cs

[10] *Ipsec.conf* [online]. [cit. 2014-03-02]. Dostupné z: http://wiki.strongswan.org/projects/strongswan/wiki/ConnSection#General-Connection-Parameters

[11] *Ipsec.secrets* [online]. [cit. 2014-03-02]. Dostupné z: http://wiki.strongswan.org/projects/strongswan/wiki/IpsecSecrets

[12] *Strongswan.conf* [online]. [cit. 2014-03-02]. Dostupné z: http://wiki.strongswan.org/projects/strongswan/wiki/StrongswanConf

[13] *StrongSwan - historie a vývoj* [online]. [cit. 2014-03-03]. Dostupné z: http://www.strongswan.org/docs/dfn_berlin_2011.pd

[14] *Aladdin eToken Pro 32k* [online]. [cit. 2014-03-05]. Dostupné z: http://www.safenet-inc.com/products/data-protection/two-factor-authentication/etoken-pro/?aldn=true

[15] *StrongSwan VPN klient* [online]. [cit. 2014-03-07]. Dostupné z: https://play.google.com/store/apps/details?id=org.strongswan.android

[16] Aplikace PKI klient. [online]. [cit. 2014-04-30]. Dostupné z: http://www.isecurity.info/downloads.aspx